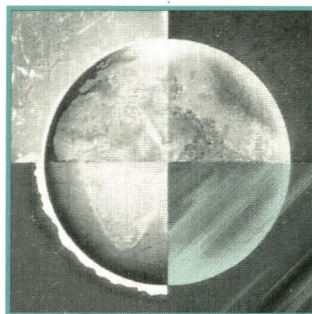


CICS Programmer's Toolkit

Installation and Administration Guide



CICS Programmer's Toolkit

Installation and Administration Guide

Version 2.0

August 1994

Document Number 200801-024030-200100

**Interlink Computer Sciences, Inc.
47370 Fremont Boulevard
Fremont, California 94538
U.S.A.
(510) 657-9800**



IMPORTANT NOTICE

Second Edition (August 1994)

The information in this document applies to the Interlink CICS Programmer's Toolkit™ Version 2.0 software. If you would like additional copies of this or any Interlink documentation, contact the Customer Support Department at our corporate address.

Copyright

Copyright © 1993, 1994 Interlink Computer Sciences, Inc. All rights reserved.

This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from the copyright holder.

Disclaimer

All information in this document is subject to periodic change and revision.

While every effort has been made to ensure the information presented in this document is accurate, Interlink disclaims liability for any inaccuracies or omissions that may have occurred.

If you find information in this document that is incorrect, misleading, or incomplete, we would appreciate your comments and suggestions. Please use the reader comment form at the back of this manual to let us know how we may improve the information we provide you.

Trademarks

This Interlink document often refers to numerous hardware and software products by their trade names. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies.

Contents

Preface

Audience	2
Organization	2
Additional Information	2
Command Notation	3

Chapter One

OVERVIEW

The Components	1-3
Programming Interface Facilities	1-4
CPT Tools	1-5
CPT API	1-8
The CPT Environment	1-9
CPT Installation	1-9
CPT Customization	1-9
CPT Initialization	1-10
CPT Termination	1-10

Chapter Two

CPT TOOLS

The Tools	2-3
Automated Transactions	2-3
Tools Customization	2-3
The LISTEN Tool	2-4
Diagnostics	2-4
The RECEIVE Tool	2-5
Customization	2-5
Reliability Factors	2-6
The SEND Tool	2-7
Customization	2-7
Reliability Factors	2-8

Chapter Three

INSTALLATION AND CONFIGURATION

Installation Overview	3-3
System/Software Requirements	3-3
Software Installation on the IBM Host	3-3
Installation Steps	3-4
SMP/E Installation	3-4
Load the CNTL File from Tape	3-4
Verify TCPNAMES is in a System CLIST Library	3-5
Allocate CPT Data Sets	3-5
Receive, Apply, and Accept the Product	3-7
Customizing the CPT Macros	3-9
Distributed Configuration File	3-9
Associated CICS Resource Definitions	3-13
CICS PLT Entries	3-14
CICS DCT Entries	3-14
CICS PPT Entries	3-16
CICS PCT Entries	3-21
SIT Entries	3-28
CPT Installation Verification Procedure (IVP)	3-29
Execution	3-29
Installing CICS TD UNIX Test Programs Using tar	3-31

200
24030-200100

Transferring Files to Your Workstation	3-31
Building the Programs	3-33
Testing the Programs	3-34
Installing the man Pages	3-34
Installing CICS TD UNIX Programs using FTP	3-36
Customizing the Translation Table	3-39

Chapter Four

ADMINISTRATOR INTERFACE

Overview	4-3
Utilization	4-7
Utilization Summary Panel	4-7
Client/Server Browse Detail Panel	4-12
Online Tracing	4-20
Online Trace Options Panel	4-21
The Online Trace Panel	4-24
The Generate Trace Entry Panel	4-26
Reset Trace Online Queue Panel	4-28
Configuration	4-30
Configuration Table Parameters Panel	4-30
Listen/Receive Tool Panel	4-35
Add/Alter Listen Tool Panel	4-40
Listen Tool API Trace Options	4-44
Send Tool Panel	4-45
Add/Alter Send Tool Panel	4-49
Send Tool API Trace Options	4-53
Operations	4-54
Operations Control Menu Panel	4-54
CPT Interface Error Log Panel	4-56
Quiesce Idle Listeners Function	4-57
Drain CPT Interface Function	4-58
Shutdown All Immediate Function	4-58
Reset All Statistics Function	4-58
Reset Online Trace Queue Function	4-59
Reset Error Log Queue Function	4-59
Startup CPT Interface Function	4-61

Chapter Five

CPT Operations

Overview	5-3
CPT Initialization	5-4
CPT Initialization Messages	5-4
CPT Initialization Failure	5-5
CPT Termination	5-6
CPT Termination Messages	5-7
TCP Transport Provider Failure	5-7
CPT Transaction Security	5-8
CPT Cumulative Statistics Records	5-9

Appendix A

CONFIGURATION STATEMENTS

Overview	A-3
CPT Definition Statements	A-4
T09MCICS	A-4
T09MLSTN	A-6
Client-Data Listener Option	A-9
T09MRECV	A-11
T09MSEND	A-14
T09MEND	A-18

Appendix B

CPT MANUAL PAGES

Man Pages Programs	B-3
CL1 (1)	B-4
SV1 (1)	B-6
UDPCLT (1)	B-8
UDPSVR (1)	B-9

Appendix C

MESSAGES AND CODES

Return Code Cross Reference Table	C-3
Messages	C-4
ABEND CODES	C-126

Index

Glossary

Reader Comment Form

200801-024030-200100



Preface

Read this guide for detailed information about installation and administration of the CICS Programmer's Toolkit, and read the ***CICS Programmer's Toolkit Programmer's Reference*** for information about the CPT API services and support for Assembler, COBOL, C, and PL/1 high-level languages.

200801-024030-200100

Audience

This guide is intended for these audiences:

- ◆ IBM system programmers who will be installing and configuring the CICS Programmer's Toolkit, and those who will be configuring the associated CICS environment.
- ◆ UNIX system administrators who will be installing and configuring the CPT sample C client/server applications.

Organization

This guide includes these chapters and appendixes:

TITLE	DESCRIPTION
Chapter One – OVERVIEW	Describes the product's function and features. It briefly describes the two application development facilities and operating environment. It also defines important concepts and terms.
Chapter Two – CPT TOOLS	Describes the automated tools and how they interface to the respective CICS and TCP/IP facilities.
Chapter Three – INSTALLATION AND CONFIGURATION	Describes how CPT is installed with SMP/E and configures to CICS tables and resources.
Chapter Four – ADMINISTRATOR INTERFACE	Describes the CPT operator commands and transactions, and diagnostic aids.
Chapter Five – CPT OPERATIONS	Describes how to manage the operation of the CPT Interface.
Appendix A – CONFIGURATION STATEMENTS	Describes CPT configuration macro instructions.
Appendix B – MAN PAGES	Contains man pages for UNIX client/server programs.
Appendix C – MESSAGES AND CODES	Lists the messages and codes generated by the CPT product. Explanations are provided for the messages with the appropriate action to take to resolve errors.

Additional Information

Refer to the these documents for additional information:

- ◆ From Interlink Computer Sciences
 - ❖ *SNS/API Programmer's Reference Manual*
- ◆ From IBM
 - ❖ *CICS/MVS Version 2.1.2 (or higher) Application Programmer's Reference Manual*
 - ❖ *CICS/MVS Version 2.1.2 (or higher) Resource Definition Macro Manual*
 - ❖ *CICS/MVS Version 2.1.2 (or higher) Resource Definition Online Manual*

Command Notation

The command notation used in this guide follows specific rules. This table illustrates these conventions:

CONVENTION	DESCRIPTION	EXAMPLE
Initial Capitalized	Window and pop-up titles.	Use the Local Output Data Set Specifications window to specify the MVS data set.
Bold, Initial Capitalized	Menu titles, menu options, functions, and fields.	Enter your name in the User Name field, your password in the Password field, and then click on Accept . Note: Click always refers to buttons.
BOLD, ALL CAPITALS	Keyboard keys.	Press RETURN .
<i>Italic</i>	The first use of words, acronyms, and phrases included in the glossary, if a glossary is included.	The address parameter is called the <i>mask</i> .
<u>Nine point Courier Bold Underline</u>	Default values in JCL statements, console commands, and Assembler operations.	These values may be set: [<u>SYNC</u> ASYNC] [, <u>LOCAL</u> REMOTE]
Nine point Courier	System output, data entry, example code, and file names. Note: As data entry, the text appears exactly as you should enter it, including capitalization.	◆ The message SHOW MESSAGES displays. ◆ The lpd server receives the data. ◆ Set the value of this argument to TRUE
<i>Nine point Courier italic</i>	Data that you must specify or supply input for, or variable information. Note: This convention may appear in conjunction with nine point Courier bold. or nine point Courier	◆ This is the command syntax df x nnn hostname ◆ The system displays this message: Process started for ID num
[] (Square brackets)	Optional arguments or commands.	PORT = ({ ipaddress hostname } [, port]) LPDEFAULT = (' :key=value [:key=value...] ')
 (Vertical bar)	A logical <i>or</i> indicating that you can select or specify one or the other of the specified values.	
{ } (Curly braces)	Indicates a list of possible items. Generally, one item appears in the message.	
... (Ellipsis)	Indicates that an item may repeat multiple times.	

200801-024030-200100



Chapter 1

OVERVIEW

This chapter provides an overview of the *CICS Programmer's Toolkit (CPT)*. It includes these sections:

- ◆ The Components
Briefly describes the CPT components.
 - ◆ The Programming Interface Facilities
Describes the facilities for communication with a TCP/IP network, CPT tools and CPT API services.
 - ◆ The CPT Environment
Briefly describes the installation and customization that must be done to make CPT available.
-

200801-024030-200100

200
24030-200100

The Components

The *CICS Programmer's Toolkit (CPT)* is an application programming interface between CICS application programs and communication subsystems, based on open network protocols. The product contains two facilities: a set of automated CICS transactions, referred to as CPT tools, and a set of CICS application program callable subroutines, referred to as CPT API services.

The CPT tools are pre-written CICS transactions that establish connections and transmit and receive data over *Transmission Control Protocol (TCP)*.

The CPT API is an *Application Program Interface (API)* that is a set of callable subroutines. These subroutines are supported by four languages that CICS programmers can use to make connections with remote TCP or *User Datagram Protocol (UDP)* processors with a great deal of flexibility.

The CPT basic operating environment supports CPT tools and CPT API services. The CPT environment is maintained by system initialization, termination, configuration and general purpose programs. These environmental programs are required, and must be customized before any communications interface can be utilized.

This manual describes the CICS Programmer's Toolkit, including CPT tools, CPT API services, installation, customization, and operations. For information related to CICS table generation and CICS application programming, read the appropriate IBM CICS manuals.

Programming Interface Facilities

CPT offers two facilities for communication with a TCP/IP network. These facilities provide a transparent network interface using a set of automated transactions, and a set of ten subroutines that can be called by user-written applications providing an easy to use API interface to CICS.

The CPT tools facility is a transparent network interface. The facility provides automated, server-connection initialization and data transfer mechanisms. The CPT tools are pre-written CICS transactions configured to CPT. The data transfer transactions utilize CICS transient data queues for communication with user-written applications. Hence, user-written application programs simply read from, and write to, configured transient data queues.

The CPT API facility is the application interface. The facility provides application developers with the ability to directly control and manage connections to the transport provider's API communication subsystem. CPT API services require application developers to use CPT callable subroutines. The subroutines provide connection initialization, data transfer, translation, and connection termination.

The CPT programming interface facilities (CPT tools or CPT API services) are application dependent. There is no internal CPT requirement that one or the other must be used. User applications can use any combination of CPT tools or CPT API services to achieve their objective.

CPT provides *Domain Name Resolver (DNR)* and extensive logging support facilities. The DNR directory services for *Internet Protocol (IP)* host and services name resolution enhance network management and reduce application program support. Logging support enhances problem analysis and statistical information.

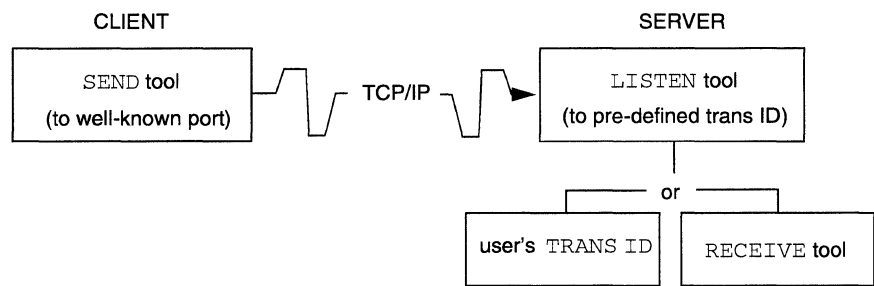
Simple UNIX code, compatible with the CPT tools data transfer transactions, is provided, allowing immediate interoperability with UNIX processors.

CPT Tools

The CPT tools include a set of automated transactions, including `LISTEN`, `RECEIVE`, and `SEND`. Together these provide a variety of processing options: translation, file or record processing, dynamic queue name resolution, dynamic host and port resolution, storage requirements, and statistics. The automated transactions can be used in combination with user-written applications, or can be used as development tools.

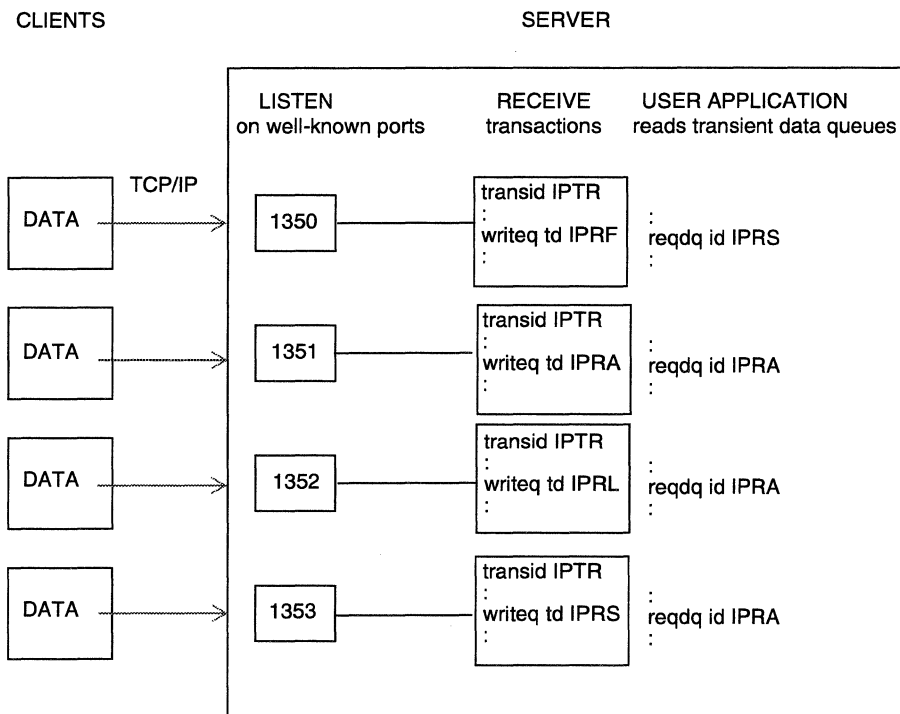
The automated `LISTEN` transaction provides connection establishment with remote client applications and initialization for data transfer transactions. The data transfer transaction can be an automatic `RECEIVE`, or a user-written application.

This figure illustrates connection-establishment using the automated `SEND` and `LISTEN` transactions; it shows how CPT tools provide automatic connection and initialization of data processing transactions.



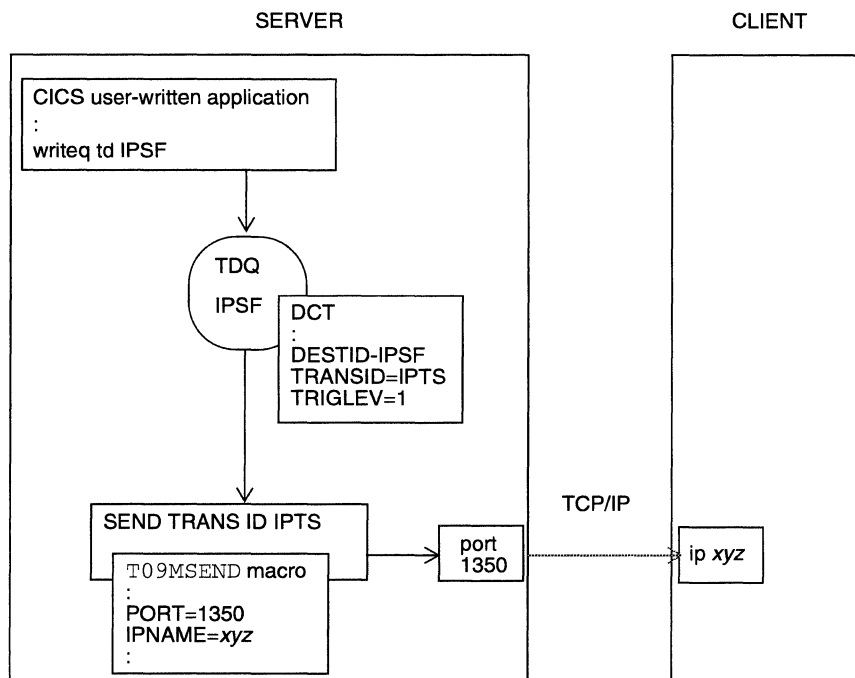
The automated `RECEIVE` transaction provides a simplex (single direction) data transfer mechanism for receiving network data. Data received from the network is parsed according to configured options and written to a transient data queue. The automated `RECEIVE` transaction is used in combination with the automated `LISTEN` transaction.

This figure illustrates the transfer of data from the client to the server CICS application. The specifics in the example are based on a sample that is shipped with the product and described in **Chapter Three – INSTALLATION AND CONFIGURATION**. It shows how `LISTEN` establishes a connection with the client on a well-known port, resulting in the creation of a `RECEIVE` transaction. `RECEIVE`'s data is written to a designated transient data queue. The user-written CICS application is triggered by the automatic transaction initialization facility of CICS when data is written to the queue.



The automated `SEND` transaction provides a simplex (single direction) data transfer mechanism for transmitting network data. Data is read from a transient data queue and parsed according to configured options before it is sent over the network.

This figure illustrates the transfer of data from a CICS application to a remote client using `SEND`. The specifics in the example are based on a sample that is shipped with the product and described in **Chapter Three – INSTALLATION AND CONFIGURATION**. It shows how a CICS application writes to the `SEND` transient data queue IPSF. When the trigger level is reached, an entry in DCT identifies the `SEND` transaction ID. The `SEND` transaction connects to the port and IP address specified in the `T09MSEND` macro defining the queue-to-host mapping.



Essentially, the two previous illustrations are valid for both the `RECEIVE` and `SEND` tools. The high-level data transfer diagram of the second figure would be reversed for `SEND`, in that all the "writes" to queues would be "reads." The more detailed information in the third figure is essentially the same for `RECEIVE`, except "reads" would be "writes," and the source of information for the `RECEIVE` transaction is `T09TMRECV`, and the information in DCT is slightly different. Read the CICS DCT table entries described in **Chapter Three – INSTALLATION AND CONFIGURATION**.

CPT API

The CPT API facility provides ten callable subroutines, or services. These callable services support connection initialization, facility management, termination, data transfer, and translation.

- ◆ TCP connection initialization involves two services: `LISTEN` allows user applications to establish server connections; `CONNECT` allows user applications to establish client connections.
- ◆ Facility management is controlled with `GIVE` and `TAKE`.
- ◆ TCP data transfer is invoked by `SEND` and `RECEIVE`.
- ◆ UDP data transfer is invoked by `SENDTO` and `RCVFROM`.
- ◆ Data translation is provided by `TRANSLATE`.
- ◆ TCP connections and UDP endpoints are terminated with `CLOSE`.

This figure shows how CPT API services permit connection, facility management, data transfer, translation, and connection release; a token passes the attributes required to perform the function of each service between the CPT environment, the user-written CICS application, and the communication subsystem.

CICS $\longleftrightarrow$ **CPT API** $\longleftrightarrow$ **TCP**

User-written Application	Connection Management LISTEN and CONNECT TOKEN (returned) API buffering tracing, statistics, connect status subtask initialization resources ownership of connection	
	Facility Management GIVE and TAKE TOKEN	
	TCP Data Transfer SEND and RECEIVE TOKEN data buffer address data buffer length	UDP Data Transfer SENDTO and RCVFROM TOKEN(Returned or Reused) API Buffering Tracing, Statistics Data Buffer Address and Length
	Translation TRANSLATE TOKEN data buffer address data buffer length translation options	
	Connection and Endpoint Release CLOSE TOKEN termination options	
		Communication Subsystem

The CPT Environment

Installation and customization must be performed before CPT is available. Installation consists of allocating the CPT data sets and executing SMP/E. Customization requires a CPT configuration file and CICS resource definitions.

The CPT configuration file defines the operating environment and automated CPT tools transactions. The environment information allows CPT to communicate with both CICS and the transport provider's API. The automated CPT tools transactions are optional, and describe data transfer characteristics.

CICS resource definitions define programs, transaction IDs, and transient data queues. Resource definition statements for CPT operating environment programs and transactions are required, while those for CPT tools transient data queues are optional.

The CICS Programmer's Toolkit can be enabled and disabled without shutting down CICS. CPT maintenance can be applied, new configurations can be loaded, or the transport provider can terminate and re-initialize without CICS being recycled.

CPT Installation

Installation and maintenance of CPT is controlled through SMP/E. CPT SMP/E target and distribution libraries need to be allocated. SMP/E receive, apply, and accept processing installs CPT from the distribution tape. Customization of the CPT configuration file and CICS resource definitions are required before the *CPT Installation Verification Procedure (IVP)* can execute.

CPT Customization

The CPT configuration file consists of required macro instructions defining the operating environment, and optional macro instructions defining the CPT tools transactions. The CPT configuration file must be assembled and link edited before the CPT initialization program is executed.

Setting up the operating environment provides these features:

- ◆ The subsystem names for communications with the transport provider
- ◆ Transaction IDs for the termination and automated listening transaction
- ◆ Log support for transient data queue names
- ◆ Default translation table

Because the CPT configuration macro references CICS programs, transaction IDs, and transient data queues, CICS resource definitions are also required.

CPT Initialization

CPT initialization is established by customizing the configuration file and executing the `T09TSTRT` program. The initialization program links the CPT environment to CICS by creating a session with the transport provider's API communication subsystem.

You can define an entry in the CICS PLT PI table to initialize CPT, or you can use the initialization program's transaction ID to set up an interactive startup. Once CPT is initialized, messages are generated to the system console indicating success or failure of the operating environment.

CPT initialization loads the configuration file and starts all of the CPT tools automated `LISTEN` services.

CPT Termination

The termination program releases all current connections, closes the session with the transport provider's API communication subsystem, and disables the CPT interface.

The CPT termination program, `T09TTERM`, can be started by defining an entry in the CICS PLT SD table, interactively, using the program's transaction ID, or with the Administrator Interface. Throughout CPT termination, messages are generated to the system console to indicate success or failure.

Chapter 2

CPT TOOLS

This chapter provides information about CPT Tools. It includes these sections:

- ◆ The Tools
Describes how to send and receive data over TCP/IP using transient data queues.
 - ◆ The `LISTEN` Tool
Describes the listening and data processing tasks of the `LISTEN` tool.
 - ◆ The `RECEIVE` Tool
Describes how the `RECEIVE` tool works, including reliability factors.
 - ◆ The `SEND` Tool
Describes the automated `SEND` tool and what it does.
-

200801-024030-200100



200-24030-200100



The Tools

The automated tools in the CICS Programmer's Toolkit are pre-written CICS code that CICS programmers can use to send and receive data over TCP/IP using transient data queues.

This feature of CPT provides immediate network capabilities with very little development effort. Application programs simply read and write to transient data queues.

Automated Transactions

CPT automated transactions provide a variety of processing options including translation, file or record processing, dynamic queue name resolution, dynamic host and port resolution, buffering requirements, and statistics. These automated transactions can be used in combination with user-written applications or can be used as development tools. These tools include:

- ◆ LISTEN
- ◆ RECEIVE
- ◆ SEND

The data flow of the automated data transfer transactions is simplex, or in a single direction. This means that the automated data transfer transactions only receive or send data. However, the automated transactions can be used in combination to achieve full-duplex data transfer, or conversational mode, to achieve a wide range of application uses.

Tools Customization

The tools are customized by CPT configuration definition statements. There is a specific macro instruction in the CPT configuration for each tool. The configuration macro instructions specify options that control tool transaction processing.

The tools are controlled by CICS table entries described in **Chapter Three – INSTALLATION AND CONFIGURATION** and CPT configuration macro statements described in **Appendix A – CONFIGURATION STATEMENTS**.

The LISTEN Tool

The LISTEN tool is selected by defining the T09MLSTN macro statement. The T09MLSTN configuration macro statement defines an automated listening transaction. The T09MLSTN macro statement contains operands that define well-known port, buffering requirements, statistics, and tracing.

Each TCP server port can be defined, along with its associated data processing transaction ID. This transaction ID can be the RECEIVE tool or a user-written program.

A LISTEN tool can also be defined to let the client determine the transaction ID to start or the transient data queue to write client specified data into.

The LISTEN tool is initiated during CPT startup processing and is defined by an entry in the CPT configuration file.

The automated LISTEN tool requires a transaction ID in order for the T09TLSTN program to be specified in the T09MCICS macro instruction. The default listener transaction ID is IPTL. The automated LISTEN tool is initiated by the T09TSTRT program during CPT initialization. A CICS START command is issued for each T09MLSTN macro instruction configured.

The automated LISTEN transaction provides a mechanism to handle connection establishment and data processing application initialization. The data processing application can be either the CPT automated RECEIVE tool or a user-written routine, providing quick development of server applications.

Diagnostics

In order to assist in application development, maintenance, support, and performance tuning, a number of diagnostic tools are provided with LISTEN. When options in the T09MLSTN macro are defined, tracing and statistics can be enabled. Traces are provided for internal flow of control, CPT API request arguments, and network data; statistics report the activity on established connections.

The RECEIVE Tool

The RECEIVE tool consists of these parts:

- ◆ An automated LISTEN transaction
- ◆ An automated RECEIVE transaction
- ◆ A user-written application.

The automated LISTEN transaction can spawn an automated RECEIVE transaction after a client connection is established. The automated RECEIVE transaction reads data from the network, processes the data, and writes records to a defined transient data queue. The user-written application is triggered when data is written to the transient data queue according to limits set in the CICS Automatic Transaction Initialization (ATI) facility. These limits are defined in the Destination Control Table (DCT) for the queue.

A single LISTEN transaction can spawn multiple, simultaneous RECEIVE data transactions, thereby servicing multiple clients simultaneously. Hence, the automated RECEIVE tool is a multi-threaded server application.

Initialization of the user application is provided by CICS when data has been written to the transient data queue. A user application simply reads records from the transient data queue. This provides a transparent network interface for CICS application developers to receive client application information.

Customization

Customization of the automated RECEIVE tool consists of both CICS resource definitions and CPT configuration statements. CICS resource definitions are required for the Processing Program Table (PPT), Program Control Table (PCT), and Destination Control Table (DCT). CPT macro definition statements are required for the automated LISTEN (T09MLSTN) and RECEIVE (T09MRECV) transactions.

The T09MLSTN macro instructions can specify server information, an automated RECEIVE transaction ID, and a reference to the T09MRECV configuration entry. The default automated RECEIVE transaction ID is IPTR. The T09MRECV macro contains operands specific to the data processing options. The data processing options specify translations, record or file control, transient data queues, statistics, and tracing. The transient data queue name can be identified directly, or dynamically resolved from the first 4 bytes received from the connection.

Reliability Factors

The `RECEIVE` tool provides a mechanism to handle data processing over the open network interface. Stream data received is collected into logical records and written to a transient data queue, which can trigger a user-written application. The idiosyncrasies associated with open network data processing is performed by the automated transaction. Additionally, the transient data queue `SYNCPOINT` and `ROLLBACK` features are used by the automated `RECEIVE` transaction to provide reliable data transfer support at the application layer.

The `RECEIVE` tool is configured with options that control when and how data is written to a transient data queue. These options describe parsing requirements:

- ◆ `FILE` specifies that data is received as a stream and is written to the transient data queue as a single record without parsing.
- ◆ `ALL`, `LL` (Logical Length), and `SEP` (Separator Errors) specify parsing, and one or more records can be written to the transient data queue.

The `RECEIVE` tool does not explicitly issue `SYNCPOINT` commands, but rather a `SYNCPOINT` command is issued by CICS during task termination. However, the CICS `ROLLBACK` facility is used within the `RECEIVE` tool when an error in processing is detected. These general categories classify errors:

- ◆ Transport provider errors
- ◆ Data processing errors

Transport Provider Errors

Transport provider errors are determined by a non-zero return code received from a CPT API service request. Typical transport provider errors are disconnect indications from a remote host and transport provider termination. Transport provider errors are recorded in the CPT error log and are referred to as the diagnostic code. Transport provider errors cause an abortive termination of the connection and cause a CICS `SYNCPOINT ROLLBACK` command to be issued. The only valid return code that is not considered an error is a release indication which is interpreted as an end-of-file notification.

Data Processing Errors

A data processing error is determined by a logic error. Typical logic errors are transient data buffer overflow, translation, logical length (`LL`), and separator (`SEP`) errors. Data processing errors are recorded in the CPT error log. Data processing errors cause an abortive termination of the connection and cause a CICS `SYNCPOINT ROLLBACK` command to be issued.

Diagnostics

A number of diagnostic tools are provided with `RECEIVE` to assist in application development, maintenance, support, and performance tuning. When options in the `T09MRECV` macro are defined, tracing and statistics can be enabled. Traces are provided for internal flow of control, CPT API request arguments, and network data; statistics report bytes processed, log maximum values, and requests from CPT API services.

The SEND Tool

The `SEND` tool consists of two programs, the automated `SEND` transaction and a user-written application. The user-written application is responsible for placing data into a transient data queue. When data is written to a transient data queue, the automated `SEND` transaction is triggered by the CICS ATI facility. These limits are defined in the DCT for the queue.

The `SEND` tool is responsible for establishing a connection and processing data. There is no restriction on the number of simultaneously executing client transactions, although you can have only one `T09MSEND` macro defined for a specific transient data queue, port, and ipname.

A user application is simply required to write data and/or records to the transient data queue. The initialization of the automated `SEND` transaction and the transfer of data is handled transparently. This provides a transparent network interface for CICS application developers to send information to server applications.

Optionally, the `SEND` tool can be initiated through its transaction ID. The command syntax would contain the transaction ID and a transient data queue name. The `SEND` tool would then process the transient data queue as if initiated by the ATI facility. This provides a restart mechanism for client applications.

Customization

Customization of the automated `SEND` tool consists of CICS resource definitions and a CPT configuration statement. CICS resource definitions are required for the user-written application Processing Program Table (PPT) and the Destination Control Table (DCT). The DCT entry for the transient data queue should specify the `T09TSEND` transaction ID. The default transaction ID for the `T09TSEND` program is `IPTS`. A CPT macro definition statement for the automated `SEND` (`T09MSEND`) transaction is required.

`T09MSEND` contains operands specific to the API transport provider and data processing options. These specify fixed or dynamic host name and port resolution, buffering requirements, statistics, and tracing. The data processing options specify translation, record or file control, transient data queue name, statistics, and tracing. The configuration options allow a user to select a fixed server address, or dynamically resolve the server address from the first transient data queue record. `T09MSEND` contains only one transient data queue name, port, and ipname. As such, multiple `T09MSEND` macros must be defined in order to address different queues, ports, or ipnames.

Reliability Factors

The SEND tool provides a mechanism to handle data processing over the open network interface. Data is read from a transient data queue and sent over the connection to a server. The idiosyncrasies associated with open network data processing is performed by the automated SEND transaction. Additionally, the transient data queue SYNCPOINT and ROLLBACK features of CICS are used by the automated SEND transaction to provide reliable transfer support.

CPT automated SEND transactions are configured with options that control how data is sent to the transport provider. These options describe control information required in the data transmission:

- ◆ FILE specifies that a connection is established and released for every record read from the transient data queue.
- ◆ ALL, LL, and SEP options specify that multiple records can be read from the transient data queue and sent over the connection. The LL and SEP options require control information to be incorporated into the output data.

The SEND tool explicitly issues SYNCPOINT commands after a connection is successfully established and released during FILE option processing. The SYNCPOINT command is not issued while processing ALL, LL, or SEP options, but rather a SYNCPOINT command is issued by CICS during task termination. However, the CICS ROLLBACK facility is used by the SEND tool when an error during processing is detected. These general categories classify errors:

- ◆ Transport provider errors
- ◆ Data processing errors

Transport Provider Error

A transport provider error is determined by a non-zero return code received from an CPT API service request. Typical transport provider errors are disconnect indications from the remote host and transport provider termination. Transport provider errors are recorded in the CPT error log and are referred to as the diagnostic code. Transport provider errors cause an abortive termination of the connection and a CICS SYNCPOINT ROLLBACK command to be issued. The only valid return code not considered an error is a release indication which is interpreted as an end-of-file notification.

Data Processing Error

A data processing error is determined by a logic error. Typical logic errors are transient data buffer overflow, translation, logical length (LL) and separator (SEP) errors. Data processing errors are recorded in the CPT error log. Data processing errors cause an abortive termination of the connection and a CICS SYNCPOINT ROLLBACK command is issued.

Diagnostics

In order to assist in application development, maintenance, support, and performance tuning, a number of diagnostic tools are provided with `SEND`. When options in the `T09MSEND` macro are defined, tracing and statistics can be enabled. Traces are provided for internal flow of control, CPT API request arguments, and network data; statistics report bytes processed, log maximum values, and requests from CPT API services.

Read **Appendix A – CONFIGURATION STATEMENTS** for information on setting up your network information and enabling these automated tools. Read **Chapter One – CPT API SERVICES** in the *CICS Programmer's Toolkit Programmer's Reference* for a description of the companion piece to the CPT Tools in the CICS Programmer's Toolkit.

200801-024030-200100



Chapter 3

INSTALLATION AND CONFIGURATION

This chapter provides information for installation and customization of the CICS Programmer's Toolkit. It includes these sections:

- ◆ Installation Overview

Briefly describes software requirements and installation on the IBM host.

- ◆ Installation Steps

Lists the contents of the installation tape and provides detailed instructions for installation of CPT.

- ◆ Customizing the CPT Macros

Describes how to customize the CPT macros, lists the contents of the configuration file distributed with the software, and describes changes you may need to make.

- ◆ Associated CICS Resource Definitions

Describes CPT definition statements that must be added to the CICS resource definition tables.

- ◆ CPT Installation Verification Procedure

Describes the *Installation Verification Procedure (IVP)*, T09TIPCK, and how to execute it.

- ◆ Installing CICS TD UNIX Test Programs using `tar`

Describes the CICS TD UNIX test programs created by the `tar` program and how to install the manual (man) pages.

- ◆ Installing CICS TD UNIX Test Programs Using FTP

Describes the FTP distribution of the CICS TD UNIX test programs.

- ◆ Customizing the Translation Table

Describes how to modify or create new translation tables using the SMP/E usermod facility.

20124030-200100

Installation Overview

The *CICS Programmer's Toolkit (CPT)* is installed on an MVS system running Interlink's SNS/TCPaccess, which must be installed prior to, or simultaneously with, the installation of CPT.

System/Software Requirements

These are the requirements for installing CPT:

- ◆ DASD space required is five cylinders
- ◆ SNS/TCPaccess 2.0 or higher
- ◆ CICS 2.1.2 or greater, or CICS 2.1.1 with IBM PTF UL60268 (PUT8908)

The CPT API service calls are compatible with these versions of software:

- ◆ VS COBOL II and above, or COBOL 370 version 3.0 and above
 - ◆ PL/I version 2.3 and above
 - ◆ Assembler release level H and above
 - ◆ SAS/C version 5.0 and above, or IBM C 1.x and above
-

Software Installation on the IBM Host

These are the requirements for installing CPT on the IBM host:

- ◆ Receive, apply, and accept the product using SMP/E
 - ◆ Receive and apply cumulative maintenance using SMP/E if supplied
 - ◆ Configure the automated transaction tools; read **Appendix A – CONFIGURATION STATEMENTS**
 - ◆ Start CPT; read **Chapter Four – ADMINISTRATOR INTERFACE**
-

Installation Steps

You will be using a standard labeled tape with VOL=SER=CPT200. The tape contains these files:

- ◆ CNTL file
- ◆ Release Notes
- ◆ SMP/E SMPMCS
- ◆ SMP/E RELFILE
- ◆ PDS UNIX code
- ◆ Compressed tar UNIX code

SMP/E Installation

Use these instructions to install SMP/E:

STEP	ACTION
1	Create and submit a JCL job stream to read the CNTL file from the distribution tape.
2	Copy the member TCPNAMES to a system CLIST library or a concatenated SYSPROC library (when you installed the SNS/TCPaccess FMID TCP0200, this CLIST was made available).
3	Use the TCPNAMES CLIST to globally edit the ALLOCT09 job to allocate the CPT data sets.
4	Use the TCPNAMES CLIST to globally edit the INSTT09 job to receive, apply and accept the CPT FMIDS feature function.

Load the CNTL File from Tape

Create a simple jobstream to read the CNTL file from the distribution tape. It will be LABEL=(1,SL). The file is a *Partitioned Data Set (PDS)* that contains all the JCL necessary to receive, apply, and accept the CPT FMIDS.



Example: This is a sample JCL to unload the CNTL file:

```
//UNLDT09      JOB
//STEP1        EXEC PGM=IEBCOPY,REGION=1024K
//INDD1         DD DSN=CNTL,VOL=SER=CPT200,UNIT=TAPUNIT,
//              LABEL=(1,SL,,EXPDT=98000)
//OUTDD1        DD DSN=TRGINDX.CNTL,DISP=SHR  --> CNTL FILE
//SYSPRINT      DD SYSOUT=*
//SYSIN         DD *
               COPY INDD=INDD1,OUTDD=OUTDD1
               /*
```

Verify TCPNAMES is in a System CLIST Library

The TCPNAMES CLIST is provided with the base product install. Ensure that this is available in a system CLIST library. This CLIST does global edits in the installation jobstreams.

Allocate CPT Data Sets

Edit the ALLOCT09 job contained in CNTL and, using the TCPNAMES CLIST, globally update the symbolics contained in the JCL. Submit this job to allocate the CPT data sets. To use TCPNAMES, execute this command on the edit command line:

```
TCPNAMES HLQ VOLSER UNIT
```

PARAMETER	DESCRIPTION
HLQ	The high-level qualifier for CPT data sets
VOLSER	The volume serial for CPT data sets
UNIT	The DASD unit type



Example: JCL to allocate CPT data sets:

```
//ALLOCT09 JOB
//*
//*  THIS JOB WILL ALLOCATE THE DATASETS REQUIRED FOR
//*  CICS PROGRAMMER'S TOOLKIT (CPT) AND WILL THEN INVOKE
//*  SMP/E TO CREATE THE DD DEFINITIONS (DDDEF'S) FOR CPT.
//*
//*  THIS JOB >> MUST << BE GLOBALLY EDITED BEFORE
//*  SUBMITTING FOR EXECUTION.  GLOBALLY CHANGE THE
//*  FOLLOWING STRINGS TO YOUR SITE-SPECIFIC DATA SET NAME
//*  HIGH LEVEL QUALIFIERS, UNIT TYPES, VOLUME NAMES, AND
//*  SYSOUT CLASS.
//*
//*  IF YOU ARE USING THE ISPF EDITOR, THE FORMAT OF THE
//*  GLOBAL CHANGE COMMAND IS:
//*      C 'OLD-STRING' 'NEW-STRING' ALL
//*
//*      'HOLDCL'          <SYSOUT HOLD CLASS
//*      'DSTINDX'         <DISTRIBUTION ZONE HIGH LEVEL QUALIFIER
//*      'DSTUNIT'         <DISTRIBUTION ZONE UNIT TYPE
//*      'DSTVOL'          <DISTRIBUTION ZONE VOLUME SERIAL NUMBER
//*      'SMPINDX'         <SMP DATASETS HIGH LEVEL QUALIFIER
//*      'TRGINDX'         <TARGET ZONE HIGH LEVEL QUALIFIER
//*      'TRGUNIT'         <TARGET ZONE UNIT TYPE
//*      'TRGVOL'          <TARGET ZONE VOLUME SERIAL NUMBER
//*
//*  VERIFY THAT THE JOB CARD AND NAMING CONVENTIONS MEET
//*  YOUR SITE'S JCL REQUIREMENTS, THEN SUBMIT THIS JOB.
//*
//*=====*
```

200801-024030-200100

```

/*          ALLOCATE PARTITIONED DATASETS FOR T09 PRODUCT          *
/*=====*
/*
/*ALLOC          EXEC PGM=IEFBR14
/*
/*          T09 TARGET (SYSTEM) DATA SETS
/*
//T09LOAD      DD DSN=TRGINDX.T09LOAD,
//              DISP=(NEW,CATLG,DELETE),UNIT=TRGUNIT,
//              SPACE=(6144,(150,15,35)),VOL=SER=TRGVOL,
//              DCB=(DSORG=PO,RECFM=U,LRECL=0,BLKSIZE=6144)
/*
//T09MAC       DD DSN=TRGINDX.T09MAC,
//              DISP=(NEW,CATLG,DELETE),UNIT=TRGUNIT,
//              SPACE=(3200,(200,10,30)),VOL=SER=TRGVOL,
//              DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=3200)
/*
//T09SAMP      DD DSN=TRGINDX.T09SAMP,
//              DISP=(NEW,CATLG,DELETE),UNIT=TRGUNIT,
//              SPACE=(3200,(400,20,30)),VOL=SER=TRGVOL,
//              DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=3200)
/*
/*          T09 DISTRIBUTION (DLIB) DATA SETS
/*
//AT09LOAD     DD DSN=TRGINDX.AT09LOAD,
//              DISP=(NEW,CATLG,DELETE),UNIT=TRGUNIT,
//              SPACE=(6144,(150,15,35)),VOL=SER=TRGVOL,
//              DCB=(DSORG=PO,RECFM=U,LRECL=0,BLKSIZE=6144)
/*
//AT09MAC      DD DSN=TRGINDX.AT09MAC,
//              DISP=(NEW,CATLG,DELETE),UNIT=TRGUNIT,
//              SPACE=(3200,(200,10,30)),VOL=SER=TRGVOL,
//              DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=3200)
/*
//AT09SAMP     DD DSN=TRGINDX.AT09SAMP,
//              DISP=(NEW,CATLG,DELETE),UNIT=TRGUNIT,
//              SPACE=(3200,(400,20,30)),VOL=SER=TRGVOL,
//              DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=3200)
/*
//SMPEDDEF     EXEC PGM=GIMSMP,REGION=4096K,TIME=120,
//              PARM='CSI=SMPINDX.CSI,PROCESS=WAIT'
//SMPDOUT      DD SYSOUT=X
//SMPLOG       DD DSN=SMPINDX.SMPLOG,DISP=MOD
//SMPCNTL      DD *

SET BOUNDARY(TCPTZN) .
UCLIN .
/* T09 TARGET LIBRARIES
ADD DDDEF(T09LOAD)                      /* LOAD LIBRARY */
    DATASET(TRGINDX.T09LOAD)
    SHR.
ADD DDDEF(T09MAC)                      /* MACRO LIBRARY */
    DATASET(TRGINDX.T09MAC)
    SHR.
ADD DDDEF(T09SAMP)                      /* SAMPLE LIBRARY*/
    DATASET(TRGINDX.T09SAMP)
    SHR.
/* T09 DISTRIBUTION LIBRARIES

```

200-24030-200100

```

ADD DDDEF (AT09LOAD)                                /* LOAD LIBRARY */
  DATASET (TRGINDX.AT09LOAD)
  SHR.
ADD DDDEF (AT09MAC)                                  /* MACRO LIBRARY */
  DATASET (TRGINDX.AT09MAC)
  SHR.
ADD DDDEF (AT09SAMP)                                  /* SAMPLE LIBRARY */
  DATASET (TRGINDX.AT09SAMP)
  SHR.
ENDUCL.

SET BOUNDARY (TCPDZN) .
UCLIN .
/* T09 DISTRIBUTION LIBRARIES                        */
ADD DDDEF (AT09LOAD)                                /* LOAD LIBRARY */
  DATASET (TRGINDX.AT09LOAD)
  SHR.
ADD DDDEF (AT09MAC)                                  /* MACRO LIBRARY */
  DATASET (TRGINDX.AT09MAC)
  SHR.
ADD DDDEF (AT09SAMP)                                  /* SAMPLE LIBRARY */
  DATASET (TRGINDX.AT09SAMP)
  SHR.
ENDUCL .
/*

```

Receive, Apply, and Accept the Product

Edit the INSTT09 job contained in CNTL. Using the TCPNAMES CLIST, globally update the symbolics contained in the JCL. Submit this job to receive, apply, and accept the CPT product FMIDS. Release 2.0 of CPT consists of two FMIDS; T090200 and T09A200. If you are installing CPT into a shared CSI, you should modify the APPLY and ACCEPT statements to specify these FMIDS.

To use TCPNAMES, execute this command on the edit command line:

```
TCPNAMES HLQ DVOL DUNIT TVOL TUNIT
```

PARAMETER	DEFINITION
HLQ	The high-level qualifier for CPT data sets
DVOL	The volume serial for CPT data sets
DUNIT	The DASD unit type
TVOL	The tape volume serial or CPT200 (in this case)
TUNIT	The tape unit type



Note: A condition code of 4 in this job is acceptable if it is produced by the SMP/E link edits. However, review the SMP/OUT for any other SMP/E errors or warnings.

**Example: JCL to install CPT product:**

```
//INSTT09 JOB
//*
//* THIS JOB WILL RECEIVE, APPLY AND ACCEPT ALL THE
//* PRODUCTS ON YOUR PRODUCT TAPE. TO SELECTIVELY INSTALL
//* SPECIFIC PRODUCTS FROM YOUR TAPE, CHANGE THE RECEIVE
//* STATEMENT TO ONLY SELECT THE PRODUCTS (FMIDS) YOU WANT.
//* EXAMPLE: RECEIVE S(CPT FMIDS).
//*
//* THIS JOB >> MUST << BE GLOBALLY EDITED BEFORE SUBMITTING
//* FOR EXECUTION. GLOBALLY CHANGE THE FOLLOWING STRINGS TO
//* YOUR SITE-SPECIFIC DATA SET NAME HIGH LEVEL QUALIFIERS,
//* UNIT NAMES, VOLUME NAMES, AND SYSOUT CLASS.
//*
//* IF YOU ARE USING THE ISPF EDITOR, THE FORMAT OF THE
//* GLOBAL CHANGE COMMAND IS:
//*      C 'OLD-STRING' 'NEW-STRING' ALL
//*
//*      'HOLDCL' <SYSOUT HOLD CLASS
//*      'SMPINDX' <SMP DATASETS HIGH LEVEL QUALIFIER
//*      'TAPUNIT' <TAPE UNIT
//*      'TAPVOL' <TAPE VOLUME SERIAL NUMBER
//*
//* VERIFY THAT THE JOB CARD AND NAMING CONVENTIONS MEET
//* YOUR SITE'S JCL REQUIREMENTS, THEN SUBMIT THIS JOB.
/*=====*
/* INVOKE SMP/E TO RECEIVE, APPLY AND ACCEPT PRODUCTS *
/*=====*
//SMPE      EXEC PGM=GIMSMP,REGION=4096K,TIME=960,
//          PARM='CSI=SMPINDX.CSI,PROCESS=WAIT'
//SMPPTFIN  DD DISP=SHR,VOL=(,RETAIN,SER=TAPVOL),
//          UNIT=TAPUNIT,LABEL=(4,SL),DSN=SMPMCS
//SMPHOLD   DD DUMMY
//SMPLOG    DD DSN=SMPINDX.SMPLOG,DISP=MOD
//SMPOUT    DD SYSOUT=HOLDCL
//SMPCNTL   DD *
//          SET BOUNDARY(GLOBAL) .
//          RECEIVE.
//          SET BOUNDARY(TCPTZN) .
//          APPLY FUNCTIONS.
//          SET BOUNDARY(TCPDZN) .
//          ACCEPT FUNCTIONS.
/*
```

Customizing the CPT Macros

The CPT configuration file consists of required macro instructions that define the operating environment and optional macro instructions that define the automated CPT tools facility. The CPT configuration file must be assembled and link edited before the CPT initialization program is executed.

The CPT configuration file contains the required macro instructions `T09MCICS` and `T09MEND`. The `T09MCICS` macro instructions define the basic operating environment. The environmental information includes the subsystem names for communications with the transport provider, transaction IDs, log support, transient data queue names, and the default translation table. The `T09MEND` macro instruction is used by internal configuration processing and contains no user defined fields.

The `T09MCICS` and `T09MEND` macro instructions are required for either CPT tools or CPT API services. CPT tools macro instructions are optional, but generally recommended. The CICS Programmer's Toolkit is distributed with CPT tools definition statements and they are used for the IVP. Additionally, the IVP can be used to verify CPT availability or diagnose operational problems.

CPT tools require the macro instructions `T09MLSTN`, `T09MRECV`, and `T09MSEND`. The use of the tools is optional.

The CPT API facility has no specific configuration macro instructions. The CPT API services use the operating environment required macros for execution.

Distributed Configuration File

Verify the information related to CPT operation and record the information required for CICS resource definitions.

The CPT configuration file, `T09CONFIG`, is customized to run CPT in a loopback mode. This is a sample CPT configuration file.



Example: CPT Loopback Configuration File

```

*-----*
*  CICS PROGRAMMER'S TOOLKIT (CPT) CONFIGURATION TABLE  *
*  *
*  SAMPLE CPT CONFIGURATION FILE TO RUN CPT IN A "LOOPBACK" *
*  MODE.  THIS CONFIGURATION FILE SHOULD BE USED TO VERIFY *
*  PROPER CPT INSTALLATION BEFORE ADDITIONAL CLIENT AND/OR *
*  SERVER DEFINITION STATEMENTS.  *
*  *
*  CERTAIN TAILORING OF THIS CONFIGURATION MAY BE NECESSARY *
*  BEFORE RUNNING CPT.  DETAILS ON THESE CONFIGURATION *
*  STATEMENTS CAN BE FOUND IN THE CPT INSTALLATION, *
*  CUSTOMIZATION, AND USER'S GUIDE.  *
*  *
*  1) VERIFY THE API SUBSYSTEM ID AND CHANGE IF NECESSARY.  *
*  *
*  2) VERIFY THE CPT START, STOP AND LISTEN CICS TRANSACTION *
*  IDS (IPST, IPPR AND IPTL) WITHIN THE CICS PCT TABLE.  *
*  *
*  3) VERIFY THE CPT ERROR, STATISTIC, TRANS TRANSIENT DATA *
*  QUEUE NAMES (ACER, ACST AND ACTR) WITHIN THE CICS DCT *
*  TABLE.  *

```

```

*
* 4) VERIFY THE CPT INSTALLATION VERIFICATION PROGRAM (IVP) *
* TRANSACTION ID (IPCK) WITHIN THE CICS PCT TABLE. *
*
* 5) VERIFY THE CPT SERVER (LISTENING) PORTS ARE NOT USED *
* BY EXISTING TCP/IP APPLICATIONS. IF A CHANGE TO THE *
* SERVER LISTENING PORT IS REQUIRED THEN MODIFY THE *
* CORRESPONDING CLIENT WELL-KNOWN PORT. *
*
* 6) VERIFY THE CPT SERVER (LISTENING) SUBTASK CICS *
* TRANSACTION ID (IPTR) WITHIN THE CICS PCT TABLE. *
*
* 7) VERIFY THE CPT SERVER (LISTENING) SUBTASK TRANSIENT *
* DATA QUEUE NAMES (IPRF, IPRL, IPRA AND IPRS) WITHIN *
* THE CICS DCT TABLE. *
*
* 8) VERIFY THE CPT CLIENT CICS TRANSACTION ID (IPTS) *
* WITHIN CICS PCT TABLE. *
*
* 9) VERIFY THE CPT CLIENT TRANSACTION DATA QUEUE NAMES *
* (IPSF, IPSL, IPSA AND IPSS) WITHIN THE CICS DCT TABLE. *
*
*-----*

SPACE 1
*
* DOMAIN DEFINITION
*
SPACE 1
T09MCICS TRANSID=(IPPR,IPTL), X
SSN=ACSS
SPACE 1
*
* SERVER DEFINITION
*
SPACE 1
T09MLSTN PARM=IPRF, X
PORT=1350, X
ANSID=IPTR, X
APISTAT=(CONN,TERM)
SPACE 1
T09MLSTN PARM=IPRL, X
PORT=1351, X
TRANSID=IPTR, X
APISTAT=(CONN,TERM)
SPACE 1
T09MLSTN PARM=IPRA, X
PORT=1352, X
TRANSID=IPTR, X
APISTAT=(CONN,TERM)
SPACE 1
T09MLSTN PARM=IPRS, X
PORT=1353, X
TRANSID=IPTR, X
APISTAT=(CONN,TERM)
SPACE 1
*
* SERVER RECEIVE TRANSACTION DEFINITION
*
SPACE 1
IPRF T09MRECV QNAME=IPRF, X
OPTIONS=(RAW,FILE), X

```

200
24030-200100

```

        TRNSTAT= (TERM)
        SPACE 1
IPRL   T09MRECV QNAME=IPRL,                                X
        OPTIONS= (LL) ,                                    X
        TRNSTAT= (TERM)
        SPACE 1
IPRA   T09MRECV QNAME=IPRA,                                X
        OPTIONS= (RAW,ALL) ,                                X
        TRNSTAT= (TERM)
        SPACE 1
IPRS   T09MRECV QNAME=IPRS,                                X
        OPTIONS= (RAW,SEP) ,                                X
        TRNSTAT= (TERM)
*
*   CLIENT DEFINITION
*
        SPACE 1
        T09MSEND QNAME=IPSF,                                X
        PORT=1350,                                           X
        IPNAME=127.0.0.1,                                    X
        OPTIONS= (RAW,FILE) ,                                X
        TRNSTAT=TERM,                                        X
        APISTAT= (CONN,TERM)
        SPACE 1
        T09MSEND QNAME=IPSL,                                X
        PORT=1351,                                           X
        IPNAME=127.0.0.1,                                    X
        OPTIONS= (LL) ,                                    X
        TRNSTAT=TERM,                                        X
        APISTAT= (CONN,TERM)
        SPACE 1
        T09MSEND QNAME=IPSA,                                X
        PORT=1352,                                           X
        IPNAME=127.0.0.1,                                    X
        OPTIONS= (RAW,ALL) ,                                X
        TRNSTAT=TERM,                                        X
        APISTAT= (CONN,TERM)
        SPACE 1
        T09MSEND QNAME=IPSS,                                X
        PORT=1353,                                           X
        IPNAME=127.0.0.1,                                    X
        OPTIONS= (RAW,SEP) ,                                X
        TRNSTAT=TERM,                                        X
        APISTAT= (CONN,TERM)
        SPACE 1
*
*   CONFIGURATION END MACRO STATEMENT
*
        SPACE 1
        T09MEND
        END

```

200801-024030-200100

Modify T09CONFIG Sample Member

These changes may need to be made to the T09CONFIG sample member. If you change T09CONFIG, first copy it into a new member. This is necessary since the T09CONFIG member is under the control of SMP and may be overwritten any time maintenance is applied to CPT.



Note: The executable load module must be named T09CONFIG.

- ◆ Verify the API subsystem ID and change if necessary.
- ◆ Verify the CPT start, stop, and listen CICS transaction IDs (IPST, IPPR, and IPTL) within the CICS PCT.
- ◆ Verify the CPT error, statistics, and trace transient data queues named (ACER, ACST and ACTR) within the CICS DCT.
- ◆ Verify the CPT IVP transaction ID (IPCK) within CICS PCT.
- ◆ Verify the CPT server (listening) ports are not being used by existing TCP/IP applications. If a change to the server listening port is required, modify the corresponding client well-known port.
- ◆ Verify the CPT server (listening) subtask CICS transaction ID (IPTL) within the CICS PCT.
- ◆ Verify the CPT server (listening) subtask transient data queues (named IPRF, IPRL, IPRA and IPRS) within the CICS PCT.
- ◆ Verify the CPT client CICS transaction ID (IPTS) within the CICS PCT.
- ◆ Verify the CPT client transient data queues (named IPSF, IPSL, IPSA, and IPSS) within the CICS DCT.

Modify CICS Execution JCL

Concatenate the CPT load library to the DFHRPL DD statement within the CICS execution JCL. For execution, either the CPT load library must be included in the CICS DFHRPL library list, or the load modules must be copied into an existing DFHRPL library. If the CPT load library is added to the DFHRPL library list, CICS must be restarted. A restart is not required when the load modules are copied into an existing DFHRPL library.

Associated CICS Resource Definitions

CPT definition statements must be added to the CICS resource definition tables or their *Resource Definition Online (RDO)* equivalents. Program and transaction ID definition statements are required, while destination and PLT definition statements are recommended, but optional. Additionally, there are a variety of resource definition statements associated with the automated CPT tools and the sample programs that can be defined to CICS.

CPT program names are required to be defined to CICS. These programs consist of system initialization, termination, *Task-Related User Exits (TRUEs)*, and configuration. These programs have required transaction IDs that must be defined. The transaction IDs are distributed with default values, but can be modified during CPT installation or customization.

The CPT initialization program can be defined in the PLTPI table. This is not required, but is generally recommended. Alternatively, the CPT initialization program can be started interactively by its transaction ID. The CPT termination program must be defined in the PLTSD table. The termination program provides proper release of transport provider connections and CPT resources during CICS shutdown. The CPT termination and initialization program transaction IDs can be used interactively to control CPT.

CPT utilizes transient data queues for logging support and for the CPT tools facility. CPT logs informational errors, statistics, and traces messages to transient data queues. These transient data queues can be existing or new queues. If the transient data queues are new, they must be defined with DCT entries. The CPT tools facility utilizes transient data queues. The facility is not required, but is recommended for IVPs and for CPT support. This requires the definition of all CPT tools transient data queues.

CPT definition statements must be in the appropriate tables either by configuring them, or by making them dynamically available through RDO commands. CICS may need to be restarted depending on your use of the installation methods described above.

CICS PLT Entries

The initialization routine resource definition shown here can be included in the PLTPI. Depending upon the release, the CPT initialization routine must appear after the entry statement for DFHDELIM or during the third stage of initialization. The resource definition statement for the T09TSTRT program is optional, but is generally recommended.

A default PLTPI table entry can be copied from the CPT SAMP library member T09PLTPI.

```
DFHPLT TYPE=ENTRY,          CICS TOOLKIT PHASE 3 / START UP      X
      PROGRAM=T09TSTRT
*
```

This termination routine resource definition must be included in the shutdown section of the PLTSD. This program provides proper termination of the CPT Interface during CICS shutdown. This includes the release of all transport provider connections and disabling the CPT Interface.

A default PLTSD table entry can be copied from the CPT SAMP library member T09PLTSD.

```
DFHPLT TYPE=ENTRY,                      X
      PROGRAM=T09TTERM
*
```



Note: If you create PLTPI and PLTSD entries from these samples only, you must add **TYPE=INITIAL** and **TYPE=FINAL** statements.

CICS DCT Entries

Resource definition statements must be defined for all CPT transient data queue names in the T09CONFIG file. CPT utilizes transient queues for logging support and for the CPT tools facility. The log transient data queues are defined in the T09MCICS macro instruction, while the CPT tools transient data queues are defined in the T09MRECV and T09MSEND macro instructions.

The transient queues are configured with default names. These names can be modified, but the corresponding name must be changed in the default CPT configuration file.

Additionally, the CPT tools client (T09MSEND) configuration statement requires that the T09TSEND program be triggered. Verify that the default transaction ID name in the DCT entry selects the T09TSEND program as specified in the PCT definition statements.

Default DCT entries can be copied from the CPT SAMP library member T09DCT.

```
*  THESE DCT ENTRIES ARE FOR THE ERROR, STATISTIC AND      *
*  TRACE FACILITY.  THE INDIRECT OPTION IS SELECTED USING  *
*  THE 'CSML' TRANSIENT DATA QUEUE.  AN INSTALLATION SHOULD *
*  VERIFY THE 'CSML' DCT ENTRY OR SELECT AN ALTERNATIVE.   *
*  THESE ENTRIES ARE REQUIRED.                               *
*                                                           *
DFHDCT TYPE=INDIRECT,          ERROR LOG      X
      DESTID=ACER,                      X
```

INDDEST=CSML		
DFHDCT TYPE=INDIRECT,	STATS LOG	X
DESTID=ACST,		X
INDDEST=CSML		
DFHDCT TYPE=INDIRECT,	TRACE LOG	X
DESTID=ACTR,		X
INDDEST=CSML		
* *		
* THESE DCT ENTRIES ARE FOR THE INSTALLATION VERIFICATION *		
* PROCEDURE (IVP). THESE ENTRIES ARE OPTIONAL. *		
* *		
DFHDCT TYPE=INTRA,	RECEIVE (SERVER)	X
DESTID=IPRF,	OPTION=FILE	X
DESTFAC=FILE,		X
DESTRCV=LG,		X
TRIGLEV=0		
DFHDCT TYPE=INTRA,	RECEIVE (SERVER)	X
DESTID=IPRL,	OPTION=LL	X
DESTRCV=LG,		X
DESTFAC=FILE,		X
TRIGLEV=0		
DFHDCT TYPE=INTRA,	RECEIVE (SERVER)	X
DESTID=IPRA,	OPTION=ALL	X
DESTRCV=LG,		X
DESTFAC=FILE,		X
TRIGLEV=0		
DFHDCT TYPE=INTRA,	RECEIVE (SERVER)	X
DESTID=IPRS,	OPTION=SEP (CR/FL)	X
DESTRCV=LG,		X
DESTFAC=FILE,		X
TRIGLEV=0		
DFHDCT TYPE=INTRA,	SEND (CLIENT)	X
DESTID=IPSF,	OPTION=FILE	X
DESTRCV=LG,		X
DESTFAC=FILE,		X
TRANSID=IPTS,		X
TRIGLEV=1		
DFHDCT TYPE=INTRA,	SEND (CLIENT)	X
DESTID=IPSL,	OPTION=LL	X
DESTRCV=LG,		X
DESTFAC=FILE,		X
TRANSID=IPTS,		X
TRIGLEV=1		
DFHDCT TYPE=INTRA,	SEND (CLIENT)	X
DESTID=IPSA,	OPTION=ALL	X
DESTRCV=LG,		X
DESTFAC=FILE,		X
TRANSID=IPTS,		X
TRIGLEV=1		
DFHDCT TYPE=INTRA,	SEND (CLIENT)	X
DESTID=IPSS	OPTION=SEP (CR/FL)	X
DESTRCV=LG,		X
DESTFAC=FILE,		X
TRANSID=IPTS,		X
TRIGLEV=1		
* *		

200801-024030-200100

CICS PPT Entries

Processing program statements must be defined for all CPT base product programs. These programs are system initialization, termination, TRUE routines, configuration, translation table, and support routines. These programs can be defined through table entries or RDO. All CPT base product programs are written in assembler. The CPT base product program names cannot be modified.

Optionally, there are a number of high-level sample programs distributed that can be defined. The sample program names can be modified according to your site's requirements. Default PPT entries for the CPT base product can be copied from the CPT SAMP library member T09PPT.

T09COMON	DFHPPT TYPE=ENTRY,	CICS TOOLKIT COMMON INTERFACE	X
	PROGRAM=T09COMON		
*			
T09CONFG	DFHPPT TYPE=ENTRY,	CICS TOOLKIT CONFIGURATION	X
	PROGRAM=T09CONFG		
*			
T09TSTRT	DFHPPT TYPE=ENTRY,	CICS TOOLKIT START-UP PROGRAM	X
	PROGRAM=T09TSTRT		
*			
T09TTERM	DFHPPT TYPE=ENTRY,	CICS TOOLKIT TERMINATION PROGRAM	X
	PROGRAM=T09TTERM		
*			
T09TLSTN	DFHPPT TYPE=ENTRY,	CICS TOOLKIT LISTEN PROGRAM	X
	PROGRAM=T09TLSTN		
*			
T09TREC	DFHPPT TYPE=ENTRY,	CICS TOOLKIT RECEIVE PROGRAM	X
	PROGRAM=T09TREC		
*			
T09TSEND	DFHPPT TYPE=ENTRY,	CICS TOOLKIT SEND PROGRAM	X
	PROGRAM=T09TSEND		
*			
T09TIPCK	DFHPPT TYPE=ENTRY,	IVP PROGRAM	X
	PROGRAM=T09TIPCK		
*			
T09XENG	DFHPPT TYPE=ENTRY,	CICS TOOLKIT TRANSLATION TABLE	X
	PROGRAM=T09XENG		
*			
T09IUEXT	DFHPPT TYPE=ENTRY,	OPTIONAL EXIT PROGRAM	X
	PROGRAM=T09IUEXT		
*			
T09ATADD	DFHPPT TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09ATADD		
*			
T09ABDTL	DFHPPT TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09ABDTL		
*			
T09ACNFG	DFHPPT TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09ACNFG		
*			
T09ADMGR	DFHPPT TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09ADMGR		
*			
T09AEXIT	DFHPPT TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AEXIT		
*			
T09AGENT	DFHPPT TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AGENT		
*			
T09AOLWT	DFHPPT TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AOLWT		
*			
T09AHELP	DFHPPT TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AHELP		

```

*
T09ATLST DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09ATLST
*
T09AMAIN DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09AMAIN
*
T09AMENU DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09AMENU
*
T09AOCTL DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09AOCTL
*
T09APING DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09APING
*
T09AQCLS DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09AQCLS
*
T09ARSTQ DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09ARSTQ
*
T09ATSND DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09ATSND
*
T09ATROP DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09ATROP
*
T09AUTIL DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09AUTIL
*
T09AWTCH DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09AWTCH
*
T09AYANK DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09AYANK
*
T09AZAPS DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09AZAPS
*
T09MAPS  DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09MAPS
*
T09MAPT  DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09MAPT
*
T09MAPU  DFHPPT TYPE=ENTRY,    ADMINISTRATOR INTERFACE      X
          PROGRAM=T09MAPU
*

```

Use RDO and the information in this table of CPT base product program definitions to define CPT base product programs:

PROGRAM	LANGUAGE	DESCRIPTION
T09COMON	Assembler	TRUE Routines
T09CONFIG	Assembler	Configuration Table
T09TSTRT	Assembler	Initialization Transaction
T09TTERM	Assembler	Termination Transaction
T09TLSTN	Assembler	Automated Listening Transaction
T09TRECVR	Assembler	Automated Receive Transaction
T09TSEND	Assembler	Automated Send Transaction

PROGRAM	LANGUAGE	DESCRIPTION
T09TIPCK	Assembler	IVP Transaction
T09XENG	Assembler	Translation Table
T09IUEXT	Assembler	Optional Exit Program
T09ATADD	Assembler	Add/Alter Tool
T09ABDTL	Assembler	Browse Detail – Clients or Servers
T09ACNFG	Assembler	Configuration Summary
T09ADMGR	Assembler	Screen Manager
T09AEXIT	Assembler	Exit Interface
T09AGENT	Assembler	Generate Trace
T09AOLWT	Assembler	Online Queue Write
T09AHELP	Assembler	Help Presentation
T09ATLST	Assembler	Listen Tool Display
T09AMAIN	Assembler	Main Entry Manager
T09AMENU	Assembler	Main Menu Display
T09AOCTL	Assembler	Operations Menu
T09APING	Assembler	Ping Remote Host
T09ARSTQ	Assembler	Reset Trace/Error Queues
T09ATSND	Assembler	Send Tool Display
T09ATROP	Assembler	Add/Alter Trace Options
T09AUTIL	Assembler	Utilization Summary
T09AWTCH	Assembler	Online Queue Display
T09AYANK	Assembler	Online Queue Release
T09AZAPS	Assembler	Reset Statistics
T09MAPS	Assembler	Basic Mapping Support Screens
T09MAPT	Assembler	Basic Mapping Support Screens
T09MAPU	Assembler	Basic Mapping Support Screens

Default PPT entries for the CPT sample programs can be copied from the CPT SAMP library member T09PPTSM.

```

T09PACL1 DFHPPT TYPE=ENTRY,    CPT ASSEMBLER CLIENT PGM 1      X
          PROGRAM=T09PACL1
*
T09PACL2 DFHPPT TYPE=ENTRY,    CPT ASSEMBLER CLIENT PGM 2      X
          PROGRAM=T09PACL2
*
T09PASV1 DFHPPT TYPE=ENTRY,    CPT ASSEMBLER SERVER PGM 1      X
          PROGRAM=T09PASV1
*

```

T09PASV2	DFHPPT TYPE=ENTRY, PROGRAM=T09PASV2	CPT ASSEMBLER SERVER PGM 2	X
*			
T09PASV3	DFHPPT TYPE=ENTRY, PROGRAM=T09PASV3	CPT ASSEMBLER SERVER PGM 3	X
*			
T09PASV4	DFHPPT TYPE=ENTRY, PROGRAM=T09PASV4	CPT ASSEMBLER SERVER PGM 4	X
*			
T09PACLU	DFHPPT TYPE=ENTRY, PROGRAM=T09PACLU	CPT ASSEMBLER UDP CLIENT	X
*			
T09PASVU	DFHPPT TYPE=ENTRY, PROGRAM=T09PASVU	CPT ASSEMBLER UDP SERVER	X
*			
T09PCCL1	DFHPPT TYPE=ENTRY, PGMLANG=COBOL, PROGRAM=T09PCCL1	CPT COBOL CLIENT PGM 1	X X
*			
T09PCCL2	DFHPPT TYPE=ENTRY, PGMLANG=COBOL, PROGRAM=T09PCCL2	CPT COBOL CLIENT PGM 2	X X
*			
T09PCSV1	DFHPPT TYPE=ENTRY, PGMLANG=COBOL, PROGRAM=T09PCSV1	CPT COBOL SERVER PGM 1	X X
*			
T09PCSV2	DFHPPT TYPE=ENTRY, PGMLANG=COBOL, PROGRAM=T09PCSV2	CPT COBOL SERVER PGM 2	X X
*			
T09PCSV3	DFHPPT TYPE=ENTRY, PGMLANG=COBOL, PROGRAM=T09PCSV3	CPT COBOL SERVER PGM 3	X X
*			
T09PCSV4	DFHPPT TYPE=ENTRY, PGMLANG=COBOL, PROGRAM=T09PCSV4	CPT COBOL SERVER PGM 4	X X
*			
T09PCCLU	DFHPPT TYPE=ENTRY, PGMLANG=COBOL, PROGRAM=T09PCCLU	CPT COBOL UDP CLIENT	X X
*			
T09PCSVU	DFHPPT TYPE=ENTRY, PGMLANG=COBOL, PROGRAM=T09PCSVU	CPT COBOL UDP SERVER	X X
*			
T09PPCL1	DFHPPT TYPE=ENTRY, PGMLANG=PL/I, PROGRAM=T09PPCL1	CPT PL/I CLIENT PGM 1	X X
*			
T09PPCL2	DFHPPT TYPE=ENTRY, PGMLANG=PL/I, PROGRAM=T09PPCL2	CPT PL/I CLIENT PGM 2	X X
*			
T09PPSV1	DFHPPT TYPE=ENTRY, PGMLANG=PL/I, PROGRAM=T09PPSV1	CPT PL/I SERVER PGM 1	X X
*			
T09PPSV2	DFHPPT TYPE=ENTRY, PGMLANG=PL/I, PROGRAM=T09PPSV2	CPT PL/I SERVER PGM 2	X X
*			
T09PPSV3	DFHPPT TYPE=ENTRY, PGMLANG=PL/I, PROGRAM=T09PPSV3	CPT PL/I SERVER PGM 3	X X
*			
T09PPSV4	DFHPPT TYPE=ENTRY, PGMLANG=PL/I, PROGRAM=T09PPSV4	CPT PL/I SERVER PGM 4	X X
*			
T09PPCLU	DFHPPT TYPE=ENTRY,	CPT PL/I UDP CLIENT	X

2008n1-024030-200100


```

PROGRAM=T09PPCLU
*
T09PPSVU DFHPPT TYPE=ENTRY,    CPT PL/I UDP SERVER      X
          PGMLANG=PL/I,
          PROGRAM=T09PPSVU      X
*
T09PSCL1 DFHPPT TYPE=ENTRY,    CPT SAS C CLIENT PGM 1      X
          PROGRAM=T09PSCL1
*
T09PSCL2 DFHPPT TYPE=ENTRY,    CPT SAS C CLIENT PGM 2      X
          PROGRAM=T09PSCL2
*
T09PSSV1 DFHPPT TYPE=ENTRY,    CPT SAS C SERVER PGM 1      X
          PROGRAM=T09PSSV1
*
T09PSSV2 DFHPPT TYPE=ENTRY,    CPT SAS C SERVER PGM 2      X
          PROGRAM=T09PSSV2
*
T09PSSV3 DFHPPT TYPE=ENTRY,    CPT SAS C SERVER PGM 3      X
          PROGRAM=T09PSSV3
*
T09PSSV4 DFHPPT TYPE=ENTRY,    CPT SAS C SERVER PGM 4      X
          PROGRAM=T09PSSV4
*
T09PSCLU DFHPPT TYPE=ENTRY,    CPT SAS C UDP CLIENT      X
          PROGRAM=T09PSCLU
*
T09PSSVU DFHPPT TYPE=ENTRY,    CPT SAS C UDP SERVER      X
          PROGRAM=T09PSSVU
*
*

```

Use RDO and the information in this table of CPT sample program definitions to define CPT sample programs:

PROGRAM	LANGUAGE	DESCRIPTION
T09PACL1	Assembler	Assembler TCP Client 1 program
T09PACL2	Assembler	Assembler TCP Client 2 program
T09PASV1	Assembler	Assembler TCP Server 1 program
T09PASV2	Assembler	Assembler TCP Server 2 program
T09PASV3	Assembler	Assembler TCP Server 3 program
T09PASV4	Assembler	Assembler TCP Server 4 program
T09PACLU	Assembler	Assembler UDP Client program
T09PASVU	Assembler	Assembler UDP Server program
T09PCCL1	COBOL	COBOL TCP Client 1 program
T09PCCL2	COBOL	COBOL TCP Client 2 program
T09PCSV1	COBOL	COBOL TCP Server 1 program
T09PCSV2	COBOL	COBOL TCP Server 2 program
T09PCSV3	COBOL	COBOL TCP Server 3 program
T09PCSV4	COBOL	COBOL TCP Server 4 program
T09PCCLU	COBOL	COBOL UDP Client program

PROGRAM	LANGUAGE	DESCRIPTION
T09PCSVU	COBOL	COBOL UDP Server program
T09PPCL1	PL/I	PL/I TCP Client 1 program
T09PPCL2	PL/I	PL/I TCP Client 2 program
T09PPSV1	PL/I	PL/I TCP Server 1 program
T09PPSV2	PL/I	PL/I TCP Server 2 program
T09PPSV3	PL/I	PL/I TCP Server 3 program
T09PPSV4	PL/I	PL/I TCP Server 4 program
T09PPCLU	PL/I	PL/I UDP Client program
T09PPSVU	PL/I	PL/I UDP Server program
T09PSCL1	Assembler	SAS C TCP Client 1 program
T09PSCL2	Assembler	SAS C TCP Client 2 program
T09PSSV1	Assembler	SAS C TCP Server 1 program
T09PSSV2	Assembler	SAS C TCP Server 2 program
T09PSSV3	Assembler	SAS C TCP Server 3 program
T09PSSV4	Assembler	SAS C TCP Server 4 program
T09PSCLU	Assembler	SAS C UDP Client program
T09PSSVU	Assembler	SAS C UDP Server program

CICS PCT Entries

Program control statements must be defined for CPT base product transactions. These transactions consist of system initialization, termination, automated tool routines and support routines. These transaction IDs can be defined through table entries or RDO and can be modified.

The CPT base product is distributed with default transaction IDs. These transaction IDs are specified within the CPT configuration file. If the default transaction IDs are modified then the change must be reflected in the CPT configuration file.

Optionally, there are a number of high-level sample programs distributed that may require transaction IDs to be defined. According to your site's requirements, the sample program transaction IDs may be modified. Additionally, verify the program names defined in the PPT resources definition statements correspond to the PCT statements.

Default PCT entries for the CPT base product transaction IDs can be copied from the CPT SAMP library member T09PCT.

*

DFHPCT TYPE=ENTRY,	CICS TOOLKIT STARTUP	X
PROGRAM=T09TSTRT,		X
TRANSID=IPST		

DFHPCT	TYPE=ENTRY,	CICS TOOLKIT SHUTDOWN	X
	PROGRAM=T09TTERM,		X
	TRANSID=IPPR		
DFHPCT	TYPE=ENTRY,	CICS TOOLKIT LISTEN	X
	PROGRAM=T09TLSTN,		X
	TRANSID=IPTL		
DFHPCT	TYPE=ENTRY,	CICS TOOLKIT RECEIVE	X
	PROGRAM=T09TRECVR,		X
	TRANSID=IPTR		
DFHPCT	TYPE=ENTRY,	CICS TOOLKIT CLIENT (SEND)	X
	PROGRAM=T09TSEND,		X
	TRANSID=IPTS		
DFHPCT	TYPE=ENTRY,	CICS TOOLKIT IVP	X
	PROGRAM=T09TIPCK,		X
	TRANSID=IPCK		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09ATADD,		X
	TRANSID=IPAA		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09ABDTL,		X
	TRANSID=IPAB		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09ACNFG,		X
	TRANSID=IPAC		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AQCLS,		X
	TRANSID=IPAE		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AGENT,		X
	TRANSID=IPAG		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AOLWT,		X
	TRANSID=IPAI		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AHELP,		X
	TRANSID=IPAK		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09ATLST,		X
	TRANSID=IPAL		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AMAIN,		X
	TRANSID=IPAM		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AMENU,		X
	TRANSID=IPAN		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AOCTL,		X
	TRANSID=IPAO		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09APING,		X
	TRANSID=IPAP		
DFHPCT	TYPE=ENTRY,	ADMINISTRATOR INTERFACE	X
	PROGRAM=T09AQCLS,		X
	TRANSID=IPAQ		

DFHPCT	TYPE=ENTRY, PROGRAM=T09ARSTQ, TRANSID=IPAR	ADMINISTRATOR INTERFACE	X
			X
DFHPCT	TYPE=ENTRY, PROGRAM=T09ATSND, TRANSID=IPAS	ADMINISTRATOR INTERFACE	X
			X
DFHPCT	TYPE=ENTRY, PROGRAM=T09ATROP, TRANSID=IPAT	ADMINISTRATOR INTERFACE	X
			X
DFHPCT	TYPE=ENTRY, PROGRAM=T09AUTIL, TRANSID=IPAU	ADMINISTRATOR INTERFACE	X
			X
DFHPCT	TYPE=ENTRY, PROGRAM=T09AWTCH, TRANSID=IPAW	ADMINISTRATOR INTERFACE	X
			X
DFHPCT	TYPE=ENTRY, PROGRAM=T09AYANK, TRANSID=IPAY	ADMINISTRATOR INTERFACE	X
			X
DFHPCT	TYPE=ENTRY, PROGRAM=T09AZAPS, TRANSID=IPAZ	ADMINISTRATOR INTERFACE	X
			X

*

Use RDO and the information in this table of CPT base product transaction ID definitions to define CPT base product transaction IDs:

PROGRAM	TRANS ID	DESCRIPTION
T09TSTRT	IPST	Initialization Transaction
T09TTERM	IPPR	Termination Transaction
T09TLSTN	IPTL	Automated Listening Transaction
T09TREC V	IPTR	Automated Receive Transaction
T09TSEND	IPTS	Automated Send Transaction
T09TIPCK	IPCK	IVP Transaction
T09ATADD	IPAA	Add/Alter Tool
T09ABDTL	IPAB	Browse Detail – Clients or Servers
T09ACNFG	IPAC	Configuration Summary
T09AQCLS	IPAE	Close Connections
T09AGENT	IPAG	Generate Trace
T09AOLWT	IPAI	Online Queue Write
T09AHELP	IPAK	Help Presentation
T09ATLST	IPAL	Listen Tool Display
T09AMAIN	IPAM	Main Entry Manager
T09AMENU	IPAN	Main Menu Display
T09AOCTL	IPAO	Operations Menu
T09APING	IPAP	Ping Remote Host

PROGRAM	TRANS ID	DESCRIPTION
T09AQCLS	IPAQ	Close Connections
T09ARSTQ	IPAR	Reset Trace/Error Queues
T09ATSND	IPAS	Send Tool Display
T09ATROP	IPAT	Add/Alter Trace Options
T09AUTIL	IPAU	Utilization Summary
T09AWTCH	IPAW	Online Queue Display
T09AYANK	IPAY	Online Queue Release
T09AZAPS	IPAZ	Reset Statistics

Default PCT entries for the CPT sample program transaction IDs can be copied from the CPT SAMP library member T09PCTSM.

```

*
DFHPCT TYPE=ENTRY,      SAMPLE ASSEMBLER CLIENT PGM 1      X
      PROGRAM=T09PACL1,
      TRANSID=IAC1
*
DFHPCT TYPE=ENTRY,      SAMPLE ASSEMBLER CLIENT PGM 2      X
      PROGRAM=T09PACL2,
      TRANSID=IAC2
*
DFHPCT TYPE=ENTRY,      SAMPLE ASSEMBLER SERVER PGM 1      X
      PROGRAM=T09PASV1,
      TRANSID=IAS1
*
DFHPCT TYPE=ENTRY,      SAMPLE ASSEMBLER SERVER PGM 2      X
      PROGRAM=T09PASV2,
      TRANSID=IAS2
*
DFHPCT TYPE=ENTRY,      SAMPLE ASSEMBLER SERVER PGM 3      X
      PROGRAM=T09PASV3,
      TRANSID=IAS3
*
DFHPCT TYPE=ENTRY,      SAMPLE ASSEMBLER SERVER PGM 4      X
      PROGRAM=T09PASV4,
      TRANSID=IAS4
*
DFHPCT TYPE=ENTRY,      SAMPLE ASSEMBLER UDP CLIENT        X
      PROGRAM=T09PACLU,
      TRANSID=IACU
*
DFHPCT TYPE=ENTRY,      SAMPLE ASSEMBLER UDP SERVER        X
      PROGRAM=T09PASVU,
      TRANSID=IASU
*
DFHPCT TYPE=ENTRY,      SAMPLE COBOL CLIENT PGM 1          X
      PROGRAM=T09PCCL1,
      RSL=PUBLIC,
      TRANSID=ICC1
*
DFHPCT TYPE=ENTRY,      SAMPLE COBOL CLIENT PGM 2          X
      PROGRAM=T09PCCL2,
      RSL=PUBLIC,
      TRANSID=ICC2
*
DFHPCT TYPE=ENTRY,      SAMPLE COBOL SERVER PGM 1          X
      PROGRAM=T09PCSV1,
      RSL=PUBLIC,
      TRANSID=ICS1
*
DFHPCT TYPE=ENTRY,      SAMPLE COBOL SERVER PGM 2          X
      PROGRAM=T09PCSV2,
      RSL=PUBLIC,
      TRANSID=ICS2
*
DFHPCT TYPE=ENTRY,      SAMPLE COBOL SERVER PGM 3          X
      PROGRAM=T09PCSV3,
      RSL=PUBLIC,
      TRANSID=ICS3
*
DFHPCT TYPE=ENTRY,      SAMPLE COBOL SERVER PGM 4          X
      PROGRAM=T09PCSV4,
      RSL=PUBLIC,
      TRANSID=ICS4
*
DFHPCT TYPE=ENTRY,      SAMPLE COBOL UDP CLIENT            X
      PROGRAM=T09PCCLU,
      RSL=PUBLIC,
      TRANSID=ICCU
*

```

200801-024030-200100

	DFHPCT TYPE=ENTRY, SAMPLE COBOL UDP SERVER	X
	PROGRAM=T09PCSVU,	X
	RSL=PUBLIC,	X
	TRANSID=ICSU	
*		
	DFHPCT TYPE=ENTRY, SAMPLE PL/I CLIENT PGM 1	X
	PROGRAM=T09PPCL1,	X
	RSL=PUBLIC,	X
	TRANSID=IPC1	
*		
	DFHPCT TYPE=ENTRY, SAMPLE PL/I CLIENT PGM 2	X
	PROGRAM=T09PPCL2,	X
	RSL=PUBLIC,	X
	TRANSID=IPC2	
*		
	DFHPCT TYPE=ENTRY, SAMPLE PL/I SERVER PGM 1	X
	PROGRAM=T09PPSV1,	X
	RSL=PUBLIC,	X
	TRANSID=IPS1	
*		
	DFHPCT TYPE=ENTRY, SAMPLE PL/I SERVER PGM 2	X
	PROGRAM=T09PPSV2,	X
	RSL=PUBLIC,	X
	TRANSID=IPS2	
*		
	DFHPCT TYPE=ENTRY, SAMPLE PL/I SERVER PGM 3	X
	PROGRAM=T09PPSV3,	X
	RSL=PUBLIC,	X
	TRANSID=IPS3	
*		
	DFHPCT TYPE=ENTRY, SAMPLE PL/I SERVER PGM 4	X
	PROGRAM=T09PPSV4,	X
	RSL=PUBLIC,	X
	TRANSID=IPS4	
*		
	DFHPCT TYPE=ENTRY, SAMPLE PL/I UDP CLIENT	X
	PROGRAM=T09PPCLU,	X
	RSL=PUBLIC,	X
	TRANSID=IPCU	
*		
	DFHPCT TYPE=ENTRY, SAMPLE PL/I UDP SERVER	X
	PROGRAM=T09PPSVU,	X
	RSL=PUBLIC,	X
	TRANSID=IPSU	
*		
	DFHPCT TYPE=ENTRY, SAMPLE SAS C CLIENT PGM 1	X
	PROGRAM=T09PSCL1,	X
	TRANSID=ISC1	
*		
	DFHPCT TYPE=ENTRY, SAMPLE SAS C CLIENT PGM 2	X
	PROGRAM=T09PSCL2,	X
	TRANSID=ISC2	
*		
	DFHPCT TYPE=ENTRY, SAMPLE SAS C SERVER PGM 1	X
	PROGRAM=T09PSSV1,	X
	TRANSID=ISS1	
*		
	DFHPCT TYPE=ENTRY, SAMPLE SAS C SERVER PGM 2	X
	PROGRAM=T09PSSV2,	X
	TRANSID=ISS2	
*		
	DFHPCT TYPE=ENTRY, SAMPLE SAS C SERVER PGM 3	X
	PROGRAM=T09PSSV3,	X
	TRANSID=ISS3	
*		
	DFHPCT TYPE=ENTRY, SAMPLE SAS C SERVER PGM 4	X
	PROGRAM=T09PSSV4,	X
	TRANSID=ISS4	
*		
	DFHPCT TYPE=ENTRY, SAMPLE SAS C UDP CLIENT	X
	PROGRAM=T09PACLU,	X
	TRANSID=ISCU	

```

*
DFHPCT TYPE=ENTRY,          SAMPLE SAS C UDP SERVER      X
PROGRAM=T09PASVU,           X
TRANSID=ISSU
*

```

Use RDO and the information in this table of CPT sample program transaction ID definitions to define CPT sample program transaction IDs:

PROGRAM	TRANS-ID	DESCRIPTION
T09PACL1	IAC1	Assembler TCP Client 1 program
T09PACL2	IAC2	Assembler TCP Client 2 program
T09PASV1	IAS1	Assembler TCP Server 1 program
T09PASV2	IAS2	Assembler TCP Server 2 program
T09PASV3	IAS3	Assembler TCP Server 3 program
T09PASV4	IAS4	Assembler TCP Server 4 program
T09PACLU	IACU	Assembler UDP Client program
T09PASVU	IASU	Assembler UDP Server program
T09PCCL1	ICC1	COBOL TCP Client 1 program
T09PCCL2	ICC2	COBOL TCP Client 2 program
T09PCSV1	ICS1	COBOL TCP Server 1 program
T09PCSV2	ICS2	COBOL TCP Server 2 program
T09PCSV3	ICS3	COBOL TCP Server 3 program
T09PCSV4	ICS4	COBOL TCP Server 4 program
T09PCCLU	ICCU	COBOL UDP Client program
T09PCSVU	ICSU	COBOL UDP Server program
T09PPCL1	IPC1	PL/I TCP Client 1 program
T09PPCL2	IPC2	PL/I TCP Client 2 program
T09PPSV1	IPS1	PL/I TCP Server 1 program
T09PPSV2	IPS2	PL/I TCP Server 2 program
T09PPSV3	IPS3	PL/I TCP Server 3 program
T09PPSV4	IPS4	PL/I TCP Server 4 program
T09PSCL1	ISC1	SAS C TCP Client 1 program
T09PSCL2	ISC2	SAS C TCP Client 2 program
T09PSSV1	ISS1	SAS C TCP Server 1 program
T09PSSV2	ISS2	SAS C TCP Server 2 program
T09PSSV3	ISS3	SAS C TCP Server 3 program

PROGRAM	TRANS-ID	DESCRIPTION
T09PSSV4	ISS4	SAS C TCP Server 4 program
T09PSCLU	ISCU	SAS C UDP Client program
T09PSSVU	ISSU	SAS C UDP Server program

SIT Entries

During CPT installation or customization, pay close attention to the maximum active tasks (AMXT) and maximum number of concurrent tasks (MXT) values. The CPT server application is designed to be a long-running CICS transaction. Additionally, a server can be written to start any number of concurrent data processing tasks, the automated listening transaction is such a task. Installations should, therefore, monitor active CPT applications for the number of concurrent, active tasks.

CPT Installation Verification Procedure (IVP)

The Installation Verification Procedure (IVP) is T09TIPCK. This routine has a default transaction ID of IPCK. The program is initiated by an operator command and supports optional execution parameters.

This routine utilizes the CPT tools SEND and RECEIVE transactions to test successful installation or CPT availability. This includes CPT configuration definition statements and CICS resource definitions.

Execution

Use the instructions in this section to execute IVP.

Execute CICS

To execute CICS, submit the modified CICS JCL, or run it as a started task. These messages display at the console during CPT initialization:

```
+CPT100I T09TSTRT CPT TRUE EXIT INTERFACE ENABLED
+CPT101I T09CINIT CPT ESTABLISHED API SESSION FOR SUBSYSTEM ssid
+CPT111I T09CINIT CPT DEFAULT TRANSLATION TABLE LOADED
+CPT109I T09TSTRT CPT INITIALIZATION SUCCESSFUL
```

CPT establishes server connection(s) during initialization. This process is independent of successful CPT initialization. For example, a configured server may fail to initialize due to an existing application listening on the well-known port within the transport provider; however, other configured servers will initialize and CPT will be available to user applications.

Test CPT in Loopback Mode

After CICS and CPT initialization, you can run the IVP transaction. Enter the IVP CICS transaction ID (IPCK) from a console or terminal session connected to CICS. If the default write and read transient data queues were modified in the T09CONFIG file, those names are required on the transaction input stream.

Issue this CICS transaction ID:

IPCK writeq readq

PARAMETER	DEFINITION
writeq	The transient data queue name to which the IVP program will write.
readq	The transient data queue name from which the IVP program will read.

Upon completion of the IVP transaction, a message displays on the console or terminal indicating success or failure. This transaction may take up to 30 seconds to complete.

This message displays upon successful completion of the IVP transaction:

```
+CPT699I CPT IVP LOOPBACK TEST WAS SUCCESSFUL!!!
```

If the IVP transaction is unsuccessful, a message written to the console or terminal indicates a likely reason for the failure. Review the detailed description of the message and check the error log for additional information.

Stop CPT

Stop CPT via the termination transaction ID (IPPR). Read **Chapter Four – ADMINISTRATOR INTERFACE** for a detailed description of this command. One terminate command is sufficient to stop CPT. Additionally, CPT is stopped during CICS shutdown.

These messages display at the MVS console and the CPT109I message displays to a terminal that initiated the termination transaction:

```
+CPT103I T09CTERM CPT TERMINATED API SUBSYSTEM SESSION
+CPT132E T09TTERM CPT TRUE EXIT INTERFACE DISABLED
+CPT120I T09TTERM CPT TERMINATION SUCCESSFUL
```

Installing CICS TD UNIX Test Programs Using tar

The `tar` distribution of the CICS TD UNIX test programs is a file created by the `tar` program and then compressed using the `compress` program. The file name of the distribution file ends in `tar.Z`, indicating that it is a `tar` file and is compressed.

In the examples in this section, all user input is bold and all UNIX responses, prompts, and messages are indented in regular type following the user input.

- ◆ Create a UNIX directory to be used to copy and build the CICS TD UNIX programs (for these examples, we use a directory name of `cics`):

```
mkdir cics
```

- ◆ Change into the newly created directory:

```
cd cics
```

- ◆ Verify the working directory:

```
pwd
.../cics
```

- ◆ The `tar` distribution is shipped as a separate file on the CPT distribution tape. The data set name of the tape data set is `CICS.TAR.Z` and the DCB characteristics of the data set are `RECFM(VB)`, `LRECL(255)`, `BLKSIZE(6160)`, and `DSORG(PS)`. If the distribution file is not already downloaded to MVS, use the `IEBGENER` utility to copy the file from tape to disk:

```
//jobname JOB ...
//COPYTARZ EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=OLD,DSN=CICS.TAR.Z,UNIT=TAPE,
// LABEL=(12,SL,,EXPDT=98000),VOL=SER=CPT200
//SYSUT2 DD DISP=(,CATLG),DSN=mvs_tarz_name,UNIT=SYSDA,
// SPACE=(CYL,(1,1))
```

Transferring Files to Your Workstation

- ◆ Copy the file to UNIX using FTP from the UNIX machine (for these examples, we name the UNIX copy of the compressed `tar` distribution `cics.tar.Z`):

```
ftp mvs_host_name
```

```
Connected to mvs_host_name.
220 mvs_host_name -- FTP Server, Enter command or HELP
Name (mvs_host_name:unix_userid): mvs_userid
331 Enter PASS command
Password:mvs_password
230 Logged in - Host mvs_host_name User mvs_userid Sess# num
ftp> binary
200 OK, Ready
ftp> get 'mvs_tarz_name' cics.tar.Z
200 OK, Ready
150-Dataset open with attributes:
Type I N Stru F Mode S Path mvs_tarz_name Volser volume
```

```

Unit unit  Dsorg PS  Recfm VB  Lrecl 255  Blksize 6160
150
226-Transfer complete
Sess# num 32949 bytes sent in 0.94 seconds (35052 bytes/s)
Path mvs_tarz_name User mvs_userid Data bytes sent 32949
Disk tracks read 1
226
local: cics.tar.Z remote: mvs_tarz_name
32949 bytes received in 0.79 seconds (41 Kbytes/s)
ftp> quit
221 Session terminated

```

- ◆ Use the UNIX `zcat` and `tar` programs to uncompress and extract, respectively, the files from the compressed `tar` distribution (you can use the two commands in a shell pipeline to avoid having to uncompress the file in place:

```
zcat cics.tar | tar -xvhf -
```

```

x Makefile, 7086 bytes, 14 tape blocks
x README, 4621 bytes, 10 tape blocks
x cl1.1, 3568 bytes, 7 tape blocks
x cl1.c, 20712 bytes, 41 tape blocks
x ftp_from_mvs.sh, 7746 bytes, 16 tape blocks
x man1 symbolic link to .
x sv1.1, 4078 bytes, 8 tape blocks
x sv1.c, 21870 bytes, 43 tape blocks
x util.c, 4262 bytes, 9tape blocks
x xlate.c, 7336 bytes, 15 tape blocks
x updc1t.1, 641 bytes, 2 tape blocks
x updc1t.c 4234 bytes, 9 tape blocks
x udpsvr.1, 496 bytes, 1 tape blocks
x updsvr.c, 3036 bytes, 6 tape blocks

```

- ◆ Verify that the files were properly created:

```

ls -l

total 77
-rw-r----- 1 uid 7086  Jul 11 06:22 Makefile
-rw-r----- 1 uid 4621  Jul 13 08:22 README
-rw-r----- 1 uid 39298 Jul 20 06:49 cics.tar.Z
-rw-r----- 1 uid 3568  Jul 11 06:22 cl1.1
-rw-r----- 1 uid 20712 Jul 11 06:22 cl1.c
-rwxr-x--- 1 uid 7746   Jul 11 06:22 ftp_from_mvs.sh
lrwxrwxrwx 1 uid 1     Jul 20 06:50 man1 -> .
-rw-r----- 1 uid 4078  Jul 11 06:22 sv1.1
-rw-r----- 1 uid 21870 Jul 11 06:22 sv1.c
-rw-r----- 1 uid 641   Jul 11 06:22 udpc1t.1
-rw-r----- 1 uid 4234  Jul 11 06:22 udpc1t.c
-rw-r----- 1 uid 496   Jul 11 06:22 udpsvr.1
-rw-r----- 1 uid 3036  Jul 11 06:22 udpsvr.c
-rw-r----- 1 uid 4262  Jul 11 06:22 util.c
-rw-r----- 1 uid 7336  Jul 11 06:22 xlate.c

```

A `README` file is included with more details the files that are in the directory and the `make` options are available. You can print this file.

Building the Programs

- ◆ Run the UNIX `make` program to build the CICS TD UNIX programs from the sources in the directory (the `all` option builds both the client and the server CICS TD programs).

If you are making these commands on a Solaris workstation, you will have to edit the `Makefile` to include the proper libraries. The `Makefile` includes the instructions to do this.

`make all`

```
cc -Dunix -target sun4 -c cl1.c
cc -Dunix -target sun4 -c util.c
cc -Dunix -target sun4 -c xlate.c
cc -Dunix  cl1.o util.o xlate.o -o cl1

cl1 (TCP client) binary has been built.

cc -Dunix -target sun4 -c sv1.c
cc -Dunix  sv1.o util.o xlate.o -o sv1

sv1 (server) binary has been built.

cc -Dunix -target sun4 -c udpclt.c
cc -Dunix  udpclt.o -o udpclt

udpclt (UDP client) binary has been built.

cc -Dunix -target sun4 -c udpsvr.c
cc -Dunix  udpsvr.o -o udpsvr

udpsvr (UDP server) binary has been built.
```

The `make` output should indicate that the client and server program binaries were successfully built.

- ◆ Review the manual pages for the client and server programs:

```
man -M . cl1
man -M . sv1
man -M . udpclt
man -M . udpsvr
```

For a printed copy of the man pages, use this command to pipe the output of the above commands into a print program at your site:

```
man -M . command_name | print_program
```

- ◆ Use this series of commands to verify operation of the TCP programs in a loopback mode:

```
sv1 -o all -f x &
cl1 -o all -f cl1
cmp cl1 x
```

This runs the server (`sv1`) in the background and directs the server to receive data into file `x` using the transfer option `all` (receive all data as is). The server is started before the client so that it is running when the client connects to it. The client is started using the same transfer mode (`-o all`) and directed to transfer file `cl1` (the client binary file). This transfers file `cl1` to file `x`. Then the UNIX `cmp` program compares

the input and output files. If `cmp` exits silently, the two files are the same.

Testing the Programs

- ◆ To test the UDP sample code, start the server with the following command:

```
udpsvr >& logname &
```

This runs the server in background and writes to the file `logname`.

Start the client with the following command, and when prompted, type in a message and press <Return>.

```
udpc1t
```

```
UDP Client> Address data is now stored in 'local' &
'remote'
```

```
UDP Client> Enter message
```

```
Hello UDP world.
```

The following will be returned to the client:

```
UDP Client> sent message
```

```
UDP Client> The returned message is: Hello UDP world.
```

```
UDP Client> Its length is : 16
```

```
UDP Client> socket closed.
```

The server will exit after the message is returned. To look at the server's output, type:

```
cat logname
```

```
UDP Server> Address is stored in 'local'
```

```
UDP Server> Waiting at port # 12345
```

```
UDP Server> length of the message is: 16
```

```
UDP Server> the message is: Hello UDP world.
```

```
UDP Server> sent the message back
```

```
UDP Server> Socket closed.
```

Installing the man Pages

The `make` file can also install the binaries and man pages into system directories. This step is optional as the programs can be run from the directory in which they were created. Before running the `make install` procedure, review the `make` file to see if the target directories for the binaries and man pages are suitable for your installation. They should be `/usr/local` for the binaries and `/usr/man/man1` for the man pages. If these are not acceptable, then you can either edit the `make` file (`Makefile`) and change the directory names in the `BINDIR` and `MANDIR` macro definitions, or override the values when `make` is run.

Please note that there are different install commands depending on the platform that you are using.

```
make install      for SunOS based systems including DEC/Ultrix
```

```
make installhp    for HP based systems
```

```
make installsvr4  for SVR4 systems including RS/6000 and Solaris
```

There is also a special call, `uninstallhp`, to remove the executable code from HP systems.

To run `make` with the default directories, or if you have changed them in the `make` file, enter these commands:



Note: These commands usually require root privileges.

```
su
Password: root_password
make install
install -s cl1 /usr/local
install -s sv1 /usr/local
install -s udpc1t /usr/local
install -s udpsvr /usr/local
install cl1.1 /usr/man/man1
install sv1.1 /usr/man/man1
install udpc1t.1 /usr/man/man1
install udpsvr.1 /usr/man/man1

cl1/sv1 and udpc1t/udpsvr binaries and man pages have been
installed.
```

`exit`

To override the directory specifications, enter this command:

```
make install BINDIR=binaries_dir MANDIR=man_pages_dir
```

In either case, the `make` execution displays messages showing the commands issued and indicating that things went successfully.

After the `make install` process, you can run `make clean` to cleanup the object and binaries from the CICS TD UNIX programs directory:

```
make clean
rm -f *.o cl1 sv1 udpc1t udpsvr

cl1/sv1 udpc1t/udpsvr binaries and object files have been
deleted.
```

The effect of an install can be undone with `make uninstall`:

```
su
Password: root_password
make uninstall
rm -f /usr/local/cl1 /usr/local/sv1
rm -f /usr/man/man1/cl1.1 /usr/man/man1/sv1.1
rm -f /usr/man/cat1/cl1.1 /usr/man/cat1/sv1.1
rm -f /usr/local/udpc1t /usr/local/udpsvr
rm -f /usr/man/man1/udpc1t.1 /usr/man/man1/udpsvr.1
rm -f /usr/man/cat1/udpc1t.1 /usr/man/cat1/udpsvr.1

cl1/sv1 and udpc1t/udpsvr binaries and man pages have been
de-installed.

exit
```


Installing CICS TD UNIX Programs using FTP

The FTP distribution of the CICS TD UNIX test programs is an MVS PDS containing a shell script and the other program sources, `make` file, man pages, etc. The shell script is first downloaded to UNIX and then run to retrieve the rest of the data. The programs are then built using `make` as in the previous examples.

In the examples in this section, all user input is bold and all UNIX responses, prompts, and messages are indented in regular type following user input.

- ◆ Create a UNIX directory to be used to copy and build the CICS TD UNIX programs (for these examples, the directory name is `cics`):

```
mkdir cics
```

- ◆ Change into the newly created directory:

```
cd cics
```

- ◆ Verify the working directory:

```
pwd  
.../cics
```

- ◆ The FTP'able CICS TD UNIX programs are shipped as an unloaded PDS on the CPT distribution tape. The data set name of the tape data set is `CICS.UNIX.DATA` and the DCB characteristics of the data set are `RECFM(FB), LRECL(80), BLKSIZE(6160), and DSORG(PO)`. If the file is not already downloaded to MVS, use the `IEBCOPY` utility to copy the unloaded PDS from tape to disk:

```
//jobname JOB ...  
//COPYPDS EXEC PGM=IEBCOPY  
//SYSIN DD DUMMY  
//SYSPRINT DD SYSOUT=*  
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))  
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))  
//TAPE DD DISP=OLD,DSN=CICS.UNIX.DATA,UNIT=TAPE,  
// LABEL=(11,SL,,EXPDT=98000),VOL=SER=CPT200  
//DISK DD DISP=(,CATLG),DSN=mvs_unixdata_name,UNIT=SYSDA,  
// SPACE=(CYL,(1,1,25))  
//SYSIN DD *  
COPY INDD=TAPE,OUTDD=DISK  
/*
```

- ◆ Copy the shell script file to UNIX using FTP from the UNIX machine (for these examples, we name the download shell script `ftp_from_mvs.sh`):

```
ftp mvs_host_name
```

```
Connected to mvs_host_name.
```

```
220 mvs_host_name -- FTP Server, Enter command or HELP
```

```
Name (mvs_host_name:unix_userid): mvs_userid
```

```
331 Enter PASS command
```

```
Password:mvs_password
```

```
230 Logged in - Host mvs_host_name User mvs_userid Sess# num
```

```
ftp> get 'mvs_unixdata_name(cptusftp)' ftp_from_mvs.sh
```

```
200 OK, Ready
```

```
150-Dataset open with attributes:
```

```
Type A N      Tabs      8      Stru F      Mode S
```

```
Path mvs_unixdata_name(CPTUSFTP) Volser volume Unit unit
```

```
Dsorg PO Recfm FB Lrecl 80 Blksize 6160
```

```
150
```

```
226-Transfer complete
```

```
Sess# num 7959 bytes sent in 2.07 seconds (3844 bytes/s)
```

```
Path mvs_unixdata_name(CPTUSFTP) User mvs_userid Data bytes  
sent 35440
```

```
Disk tracks read 2
```

```
226
```

```
local: ftp_from_mvs.sh remote: mvs_unixdata_name(cptusftp)
```

```
7959 bytes received in 1.7 seconds (4.6 Kbytes/s)
```

```
ftp> quit
```

```
221 Session terminated
```

- ◆ Make the shell script executable:

```
chmod +x ftp_from_mvs.sh
```

- ◆ Verify that the execute permission bit(s) have been set for the file (use the `-l` option):

```
ls -l ftp_from_mvs.sh
```

```
-rwxrwx--- 1 uid 7748 Oct 9 17:10 ftp_from_mvs.sh
```

- ◆ Execute the shell script and respond to the prompts. The script will FTP the rest of the CICS TD UNIX program sources, `make` file, man pages, etc. to UNIX. Enter these commands:

ftp_from_mvs.sh

```
Enter MVS data set name for CICS UNIX programs data set
(fully-qualified without quotes): mvs_unixdata_name
Enter MVS host name for FTP: mvs_host_name
Enter MVS userid for FTP: mvs_userid
Enter password for MVS userid 'cam': mvs_password
Do you want an FTP log created? (y/n): n
```

```
MVS file name (quotes added by script): 'mvs_unixdata_name'
MVS host name for FTP: mvs_host_name
MVS userid for FTP: mvs_userid
FTP log to be created: n
```

The following target files will be deleted and recreated by this process:

```
Makefile README cl1.1 cl1.c
sv1.1 sv1.c util.c xlate.c
udpc1t.1 udpc1t.c udpsvr.1 udpsvr.c
man1
```

Are these values OK? (y/n): **y**

All target files were FTP'ed or created successfully

The script indicates that the files were FTPed successfully. If the script fails, run it with the `-d` (debug) option and request that an FTP log be made. Review the debug messages and the FTP log file to determine what the problem is.



Note: The script can be re-run as many times as necessary without having to redo previous steps in this procedure.

Proceed with the building and testing described in the **Installing CICS TD UNIX Test Programs Using tar** on page 3-31.

Customizing the Translation Table

The ability to modify or create new translation tables is controlled through the SMP/E usermod facility. To incorporate a modified or new translation table into CPT, receive and apply the modules to the CPT SMP/E CSI.

Interlink recommends that sites do not modify the default or distributed translation table named T09XENG. Distributed maintenance could regress site modifications. Update and submit member T09UTRAN in your CNTL library to apply a new translation table.

Additional translation tables are in the SNS/TCPaccess SAMP library. Copy the translation table to your T09SAMP library, modify, and run the following usermod:

```
//<NAME> JOB (000), ' TRANS USER MOD'
//*
//*      SAMPLE JCL TO RECEIVE AND APPLY USERMOD TO CHANGE
//*      THE CPT TRANSLATION TABLE (T09XENG).
//*
//*      CHANGE 'XXX' TO THE CORRECT CPT FMID.
//*
//*
//SMPE      EXEC PGM=GIMSMP,REGION=4096K,TIME=960,
//          PARM='CSI=SMPINDX.CSI,PROCESS=WAIT'
//SMPHOLD   DD  DUMMY
//SMPLOG    DD  DSN=SMPINDX.SMPLOG,DISP=MOD
//SMPOUT    DD  SYSOUT=*
//SMPPTFIN  DD  *
++ USERMOD (MU1TRAN) .
++ VER (Z038)
          FMID(T090xxx) /* CHANGE TO CPT FMID */ .
++ SRC (T09XENG) TXLIB(T09SAMP)
DISTMOD(AT09LOAD)DISTLIB(AT09SAMP) .
/*
//SMPCNTL   DD  *
          SET BDY(GLOBAL) .
          RECEIVE S(MU1TRAN) .
          SET BDY(TCPTZN) .
          APPLY  S(MU1TRAN) .
/*
```



Chapter 4

ADMINISTRATOR INTERFACE

This chapter describes the CICS transactions and panels of the Administrator Interface. It includes these sections:

- ◆ Overview
A summary of the Administrator Interface facilities.
 - ◆ Utilization
Describes monitoring, tracing, and termination of TCP connections.
 - ◆ Configuration
Describes how to display, update, and add CPT Tool definitions.
 - ◆ Operations
Describes startup and graceful termination of the CPT Interface, and control of diagnostic tool and statistics.
-

200801-024030-200100

20
024030-200100

Overview

The CPT Administrator Interface is a set of CICS transactions and screens that give improved visibility and control of the TCP/IP connections being established through the CPT Interface. It provides a system administrator with the tools to trace, diagnose, and terminate any CICS transaction using TCP or UDP calls.

The Administrator Interface also allows dynamic modification of the CPT tools defined in the Configuration Table, and the ability to create new CPT tool definitions. The features of the Administrator Interface are grouped into these major components:

- ◆ Utilization – monitoring, tracing, and termination of TCP connections.
- ◆ Configuration – display, update, and addition of Configuration Table entries.
- ◆ Operations – startup and graceful termination of the CPT Interface, and control of diagnostic tools and statistics.

The following sections describe each of the CICS screens (called panels) used by the Administrator Interface, and all of the services they provide to a system administrator.

Entering the Administrator Interface

The **Main Menu** for the Administrator Interface is invoked with the IPAM transaction. This transaction produces the **Main Menu** display as shown in this example:



Note: All the Administrator Interface transactions have a three-character prefix of IPA. This default can be changed at installation if these transaction IDs conflict with existing CICS transids. (The default IPAM transaction is given here.)

```

                                WELCOME TO THE

                                INTERLINK COMPUTER SCIENCES, INC.

                                CICS PROGRAMMER'S TOOLKIT
                                ADMINISTRATOR INTERFACE

                                PF4)  U: UTILIZATION

                                PF5)  C: CONFIGURATION

                                PF6)  O: OPERATIONS

                                PF9)  K: NAVIGATION

                                >
                                Pfl=?
  
```

The **Main Menu** panel has selections for the three main entry points into the Administrator Interface: Utilization, Configuration, and Operations. In addition, there is a selection for Navigation, which describes the standard PF key and function code assignments.

Function Codes and PF Keys

Each selection on the **Main Menu** can be chosen with a PF key or by entering a function code. The function code is a one- or two-character command that can be typed on the command line at the bottom of most Administrator Interface panels.

For example, to go to the first Utilization panel from the **Main Menu**, you can either press the **PF4** function key, or type **U** on the command line and press **Enter**.

The command line in a panel is labeled by the > symbol. The command line is a 12-byte input field, followed by an error message field on the same line.

Every panel in the Administrator Interface has this combination of PF keys and function codes available to navigate from panel to panel. The PF key choices are displayed at the bottom of each panel to assist novice users, and the function codes are available for experienced users that want to jump directly to a particular panel.

The function assignments for PF keys 13 - 24 are always equivalent to the assignment of PF keys 1 - 12.

Administrator Interface Navigation

The fourth selection on the **Main Menu** brings up the online explanation of the standard PF key values and navigation function codes. This example shows this panel:

ADMINISTRATOR INTERFACE NAVIGATION				
At all times	CLEAR KEY PF1/PF13	or	"X" = or "?" =	ESCAPE to exit and return to CICS HELP to learn about the current screen
Usually	PF3/PF15 PF4/PF16 PF5/PF17		= or "U" = or "C" =	goback to return to the parent screen Utilization to see tasks Configuration to see parameters
If appropriate	PF7/PF19 PF8/PF20	or	"B" = or "F" =	Page Backward through a list Page Forward through a list

Each function in the Interface has an associated 1- or 2-byte function code. Utilization, Configuration and Operations may be entered directly from CICS by appending their 1-byte code (U/C/O) to your 3-byte system prefix, creating a CICS transaction id. In addition, a 1- or 2-byte function code may be specified as a parameter to the Main Interface program or may be input while in another function to jump over directly. Function codes are found prefixing the panel identifiers in the upper right hand corner of each display.

As this panel shows, almost every Administrator Interface panel has a brief help display that can be invoked with the **PF1** key or the '?' function code.

The **PF3** key exits a panel and return to the parent panel, the **PF8** key browses forward through a series of panels, and the **PF7** key browses backward through a series of panels.

Most of the Administrator Interface panels have a panel ID in the upper-left corner. The 2-character prefix of the panel ID is the function code for the panel.



Note: Not all panels can be jumped to directly. Some panels must have the displayed item selected directly or browsed from a previous panel.

Exiting the Administrator Interface

The **Clear** key or **X** function code can be used to exit the Administrator Interface from any panel. After exiting, this message appears, confirming the exit.

CICS PROGRAMMER'S TOOLKIT ADMINISTRATOR INTERFACE TERMINATED

Bypassing the Main Menu

In order to enter the Administrator Interface faster, you can bypass the **Main Menu** by entering the transaction ID for each major component directly from a

cleared screen. Once inside the Administrator Interface, these three major panels can always be jumped to by entering their function code or PF key.

Trans Code	Primary Panel Name for component	Func Code	PF Key
IPAU	Utilization Summary panel	U	PF4
IPAC	Configuration Table panel	C	PF5
IPAO	Operations Control Menu	O	None



Note: It is recommended that only these three transaction IDs and IPAM be used to directly enter the Administrator Interface. Using any of the other Administrator Interface transactions with the IPA prefix could have unpredictable results.

Utilization

The Utilization component of the Administrator Interface provides a real-time display of all the CICS tasks in the system that have a TCP connection or a UDP endpoint established. The utilization panels can display detail information about the current state of each connection endpoint and allow the administrator to perform several types of actions on those connections.

- ◆ The administrator can terminate any active connection by making an abortive close on the connection.
- ◆ The administrator can ping the remote host of a connection to make sure that the remote host is still alive.
- ◆ The administrator can dynamically set and remove trace options on a connection for any CPT tool or API program, and observe the trace output in the Online Trace.

These features give a CICS administrator the ability to monitor and diagnose all of the CICS transactions that are using TCP or UDP on that CICS system.

Utilization Summary Panel

The **Utilization Summary** panel is the heart of the Administrator Interface. It displays all of the endpoints that have been established through the CPT Interface. There are three types of CPT endpoints that can be displayed:

- ◆ A TCP connection endpoint
- ◆ A UDP endpoint (which is connection-less)
- ◆ A listening endpoint (which is waiting for a connection)

Each of these endpoint types can be active and owned by a CICS task, or inactive with no current task owning it. This is an example of the first page of a **Utilization Summary**.

```

04/27/94          2 UTILIZATION SUMMARY          US01
1 18:26:37 4 CPT INTERFACE STARTED AT 14:04 ON 04/27/94 IN APPLID CICSQA01 P. 1 3

5 RECV:      4089 MSGS@      146847 BYTES.      7 TASKS:      13      TOT:      19
6 SEND:      5998 MSGS@      146847 BYTES.      8 STORAGE:    282K      HWM:      282K

    TOKEN      TASKNO      TRANID      PORT      REMOTE HOST:PORT      STATE
9 000251B0          TP91      1004 TCP          INACTIVE
  00023A68      23      IPTL      3380 TCP          LISTEN-I/P
  00091A78      25      IPTL      3381 TCP          LISTEN-I/P
  00091638      27      IPTL      3383 TCP          LISTEN-I/P
  000922F8      31      TPS4      1804 TCP          LISTEN-I/P
10 00092738      33      TPC1      4364 TCP      138.70.53.17:1804      RECV---OK
  00092B78      34      TPS2      1804 TCP      138.70.53.17:4365      RECV---I/P
  00098528      35      U0S2      4365 UDP      138.70.45.12:7        RCFR---I/P
  000A1138      36      TPC2      4366 TCP      138.70.45.12:7        RECV---065
  000A1578      45      TPC1      4367 TCP      138.70.53.17:1804      RECV---I/P
  000A19B8      46      TPS2      1804 TCP      138.70.53.17:4367      RECV---I/P
  000BA5F8      50      TPC2      4368 TCP      138.70.45.12:7        RECV---OK
  00091EB8      160      IPTL      3382 TCP          LISTEN-I/P
                                     13 (MORE)

11 >          12 ..ENTER TOKEN OR TASKNO TO SEE DETAIL INFORMATION...
    Pf1=?      Pf3=goback      Pf4=refresh      Pf5=Configuration      Pf8=Fwd
14 Pf2=Inactives off      Pf6=Gen Entry      Pf10=Browse Servers      Pf11=Browse Clients

```

This table describes the screen:

ITEM	ACTION
1	The current date and time on this CICS system.
2	The panel title and panel id for this screen.
3	The current page number, if there is more than one page of data to display.
4	The date and time that the CPT Interface was initialized, and the APPLID of this CICS system.
5	The total number of CPT messages received by all CPT endpoints since CPT initialization, and their cumulative byte length.
6	The total number of CPT messages sent by all CPT endpoints since CPT initialization, and their cumulative byte length.
7	The current number of active CPT tasks in this system, and the total number of CPT tasks that have executed since CPT initialization.
8	The total amount of storage currently being used for CPT control blocks and buffers, and the high-water-mark for that storage since CPT initialization.
9	Inactive CPT endpoints, which have no active CICS task attached.
10	Active CPT endpoints, which display the TASKNO of their attached CICS task, and the host IP Address and port number of their remote partner.
11	The command line used for selecting the detail display of a particular connection, by entering its TASKNO or TOKEN address.
12	The message area for command feedback and error notification.
13	The (MORE) indicator that more panels are available as part of a browse.
14	The PF keys currently active for this panel.

200
24030-200100

Summary Statistics

The Summary Statistics shown at the top of the **Utilization Summary** panel are cumulative for all CPT transactions in this CICS system. On the left are shown the total number of CPT messages sent and received since CPT initialization and their cumulative byte length. On the right are shown the current number of active CPT tasks in the system and the total number of CPT tasks that have executed since CPT initialization. Below the CPT messages and tasks is the total amount of storage currently used for CPT control block and buffers, and the high-water-mark (HWM) for that storage since CPT initialization.



Note: If the Reset All Statistics function is performed from the **Operations Control** menu, the message and byte counts are reset to zero, and the task and storage counts are reset to their current values.

Configuration Table Statistics Parameters

In order to collect the summary statistics displayed on the **Utilization Summary** panel, two parameters must be specified in the configuration table. These parameters of the `T09MCICS` macro are described further in **Appendix A – CONFIGURATION STATEMENTS**.

- ◆ `AISTATOP=YES`
Specifies that statistics capture will be turned on.
- ◆ `AITABSZ=4096`
Specifies the number of bytes allocated to the internal table used to collect statistics and tracing requests. A 4K table can contain 80 entries.

List of CPT Endpoints

Each page of the **Utilization Summary** can display 13 CPT endpoints. The endpoint for a TCP connection can exist across CICS tasks and is passed via the `GIVE` and `TAKE` API calls. If a CICS task currently owns an endpoint, then the connection is active and the task number of the owning CICS task is displayed in the `TASKNO` column. If the endpoint has no owning CICS task at the moment, it is inactive and the `TASKNO` column is blank.

- ◆ **Active endpoints**
The screen line is highlighted; the task number is shown.
- ◆ **Inactive endpoints**
The line is not highlighted; no task number is shown.

The CPT endpoints are displayed in task number sequence, so the inactive endpoints are shown first, then the active endpoints are shown, with the oldest tasks first.

Endpoint Data Fields

Each CPT endpoint listed on the **Utilization Summary** panel has these data fields displayed in each of its columns:

◆ TOKEN

The address of the control block that represents the Connection End Point.

◆ TASKNO

The task number of the owning CICS task. This field is blank for inactive endpoints.

◆ TRANID

The transaction ID of the owning CICS task.

◆ PORT

The local port number being used, and the type of protocol being used (TCP or UDP).

◆ REMOTE HOST:PORT

The IP address and port number of the partner program on the remote host. Listener tasks have blanks in this field.

◆ STATE

The `STATE` column contains the last API call made on this endpoint, and the return code of that call. Example syntax is shown here:

API Call	Return Code
XXXXX-	RRR

❖ API Calls:

TCP API Calls: CLOSE, CONNCT, GIVE, LISTEN, RECV, SEND, TAKE.

UDP API Calls: RCFR, SNTQ.

❖ Return Codes:

I/P - In progress. The call is still being processed by the TCP transport provider.

OK - A zero return code was returned for the call.

nnn - A 3-digit, decimal return code, as specified in **Appendix C – MESSAGES AND CODES** under *Return Code Cross Reference Table*, or in a return code copy member such as `T09KCRCS` for COBOL.

Displaying Inactive Endpoints

To shorten the list of endpoints displayed on this panel, you can suppress the display of all inactive endpoints. The inactive endpoints are not currently owned by any CICS task. The **PF2** key toggles the display of these inactive endpoints on the **Utilization Summary** panel.

When inactive endpoints are being displayed, the **PF2** key is shown as:

PF2=Inactives off

When inactive endpoints are being suppressed, the **PF2** key is shown as:

PF2=Inactives

The `Inactives off` option remains in effect for only one administrative session. When first displaying the **Utilization Summary** panel the default is always `Inactives`.

Explicit Selection of the Detail Display

A detail display can be selected for any of the endpoints displayed on the **Utilization Summary** panel. To do this, **enter** either the TASKNO or TOKEN field number on the command line, as the default prompt for this panel suggests:

```
...ENTER TOKEN OR TASKNO TO SEE DETAIL INFORMATION...
```

The task number is the easiest field to enter for this selection. However, the `TOKEN` address field must be offered as an alternative for the inactive endpoints that do not currently have an owning CICS task, and thus no task number to enter.

If an invalid number is entered on the command line, then the following message appears, prompting for another TASKNO or TOKEN:

```
REQUESTED TASK NOT FOUND - PLEASE VERIFY      1234
```

Browsing Detail for Client or Server Task

These are the options for browsing through the detail displays if you do not want to select a task explicitly on the **Utilization Summary** panel:

```
PF10=Browse Servers - (Connection initiated by remote host)
```

```
PF11=Browse Clients - (Connection initiated by CICS task)
```

A Client task initiates a connection with a Server on a remote host. A Client task does not need to specify what local port it will be using, so the TCP provider automatically assigns its connection a port number, beginning with 4096 and going up.

A Server task must always keep the same port number and should not conflict with any other Server or Client port. There may be a Listener task and several subordinate Receiver tasks started by the Listener, which are all associated with the same Server port.

Generating an Online Trace Entry

You can specify online tracing options from the **Browse Detail** panel, or by using **PF6** on the **Utilization Summary** panel, as shown by this option:

```
PF6=Gen Entry
```

This option is used when you want to trace an endpoint that is not currently established. The **Browse Detail** panel can only specify tracing options for connection endpoints that currently exist. However, some connections complete so quickly, that the Online Trace entry must be defined prior to the connection initiation in order to capture it. How that trace entry is specified, and how its trace options are selected, are described further in **The Generate Trace Entry Panel** on page 4-26.

Client/Server Browse Detail Panel

The **Browse Detail** panel displays statistics for a particular endpoint and its CICS transaction. From this panel, three potential actions can be taken on the endpoint.

- ◆ End its CICS task, and close the endpoint
- ◆ Ping the remote host for the endpoint to test if it is still alive
- ◆ Set API trace options for the endpoint

The **Browse Detail** panel can display or suppress different data fields, depending on the type of CICS task and the state of the endpoint. Three examples are shown next to illustrate these different types of displays for Browse Detail.

Displaying an Active Connection Endpoint

The first example is a Server transaction waiting to receive more input on a TCP connection with a remote client. The identification line in the center of the screen (shown by item 6) displays the data fields that identify this endpoint:

- ◆ Task Number
- ◆ Local Port
- ◆ Token Address
- ◆ Protocol
- ◆ State (last API call, and Return Code)



Note: This identification is very similar to the line on the Utilization Summary for this endpoint. Refer to **Endpoint Data Fields** on page 4-9 for detailed descriptions of these fields.

The First Activity and Last Activity fields show the date and time of the first API call made on this connection, and the data and time of the most recent API call made, which is shown in the State field along with its return code. In this case, the **I/P** in the State field indicates that the Receive call is still 'In Progress'. The Server task is waiting for its last call to receive more input on the connection.

Because this endpoint has an active connection with a remote client, all the remote host information is shown, and the Endpoint Statistics are shown.

The remote host information shows the IP address of the remote host in dotted decimal notation. Following the colon is the port number being used by the Client program on the remote host. If the Domain Name Resolver is active, then the domain name of the remote host will also be displayed.

Server Transaction Detail

This figure shows a page of the **Browse Detail** panels invoked by the **Browse Servers** function. It displays one of the two current Server transactions receiving input on port 1804:

```

04/27/94          CLIENT/SERVER BROWSE DETAIL          BD01
18:34:15          2 SERVER TRANSID TPS2 ON PORT 1804          P. 6 3

+----CUMULATIVE STATISTICS FROM 04/27/94 AT 14:05:05 -----+
| RECEIVED MESSAGES:      1391      SENT MESSAGES:      1391 |
|          BYTES:          51928      BYTES:          51928 |
5 |          TOTAL CONNECTIONS:      3      CURRENT CONNECTIONS:      2 |
+-----+

6 TASK#:          34  LPORT: 1804  TOKEN: 00092B78  PROTOCOL: TCP  STATE: RECV---I/P

7 FIRST ACTIVITY@: 04/27/94 14:05:13      LAST ACTIVITY@: 04/27/94 18:33:48

8 REMOTE ADDRESS: 138.70.53.17:4365
9  HOST NAME: POGO.INTERLINK.COM.

          10 MESSAGES RECV'D:      675      SENT:      675
          BYTES RECV'D:    24738      SENT:    24738
(MORE) 12

11 >
13 Pf1=?  Pf3=goback  Pf4=Utilization  Pf5=Configuration  Pf7=Bwd  Pf8=Fwd
    Pf2=End Task (close)      Pf6=Ping Remote      Pf9=Trace Options
  
```

This table describes the panel:

ITEM	DESCRIPTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID and current page number.
4	The transaction ID of the Server task and the local port it is using.
5	The cumulative statistics being collected for all instances of the Server transactions using this port number.
6	The identification line for this TCP endpoint, giving its task number, local port number, token address, protocol, and current state.
7	The date and time of the first API call on this connection, and the date and time of the most recent API call, which is shown in the state field.
8	The IP address of the remote host in dotted decimal notation, followed by the port number being used by the client program on the remote host.
9	The domain name of the remote host as resolved by the DNR subsystem.
10	The endpoint statistics being collected for just this specific connection.
11	The command line used for specifying an Administrator Interface function code.
12	The (MORE) indicator that more panels are available as part of a browse.
13	The PF keys currently active for this panel.

Endpoint Statistics

The count fields shown below the remote host information are the statistics for this specific endpoint. The fields show the number of messages received and sent since the first activity of this endpoint, and the total byte count for the messages sent and received. These counts are rolled into the Cumulative Statistics shown at the top of the panel, above the identification line.

Cumulative Statistics

The Cumulative Statistics box at the top of the panel shows how many total and current connections have been established to CICS local port,1804. This box shows the cumulative message and byte counts for all instances of the Server transactions using this port number.

Controlling Statistics Capture

The Cumulative Statistics are collected for all CPT transactions when the `AISTATOP=YES` and `AITABSZ=` parameters are specified in the CPT Configuration Table. These parameters of the `T09MCICS` macro are described further in **Appendix A – CONFIGURATION STATEMENTS**. These parameters cannot be changed dynamically within the Administrator Interface, and must be defined prior to CPT initialization.

The counters for both the Cumulative Statistics and the Endpoint Statistics can be reset to zero from the Operations Menu in the Administrator Interface, as described in **Reset All Statistics Function** on page 4-58. The date and time shown in the Cumulative Statistics box will be the last time the statistics were reset or when the CPT Interface was initialized.

2012-10-30-200100

Listener Transaction Detail

This second example is a CPT Listener task waiting to receive new connection requests on the same local port number as the previous server example.

This figure shows a single **Browse Detail** panel that was selected directly by TASKNO from the **Utilization Summary** panel:

```

1 04/27/94          2 CLIENT/SERVER BROWSE DETAIL          BD01 3
  18:36:11          4 SERVER TRANSID TPS4 ON PORT 1804

+---CUMULATIVE STATISTICS FROM 04/27/94 AT 14:05:05 ---+
|
5 | RECEIVED MESSAGES:      1391          SENT MESSAGES:      1391 |
  | BYTES:      51928          BYTES:      51928 |
|
| TOTAL CONNECTIONS:      3          CURRENT CONNECTIONS:      2 |
+-----+

6 TASK#:      31  LPORT: 1804  TOKEN: 000922F8  PROTOCOL: TCP  STATE: LISTEN-I/P

7 >
8 Pf1=?  Pf3=goback  Pf4=Utilization  Pf5=Configuration
  Pf2=End Task (stop listen)          Pf9=Trace Options
  
```

ITEM	DESCRIPTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID.
4	The transaction ID of the Server task and the local port it is using.
5	The cumulative statistics being collected for all instances of the Server transactions using this port number.
6	The identification line for this listen endpoint, giving its task number, local port number, token address, protocol, and current state
7	The command line used for specifying an Administrator Interface function code.
8	The PF keys currently active for this panel.



Note: This panel is missing these data fields that are included in the previous Browse Detail example:

- ❖ No remote host information is shown because a Listener task has no established connection. It is just waiting for a new connection request to come in from any remote host.

- ❖ No endpoint statistics are shown because the Listener task has only a listening endpoint, and no established connection.
- ❖ No **PF6** key to ping the remote host since no connection exists.
- ❖ No **PF7** or **PF8** key since this panel is not part of a browse.
- ❖ No page number is shown since this panel is not part of a browse.

Displaying a Listener Endpoint

A Listener task has an endpoint, but no connection with a remote host. It will usually start a separate Server task when a new connection request comes in from a remote host. This is why the bottom half of the screen is empty for a Listener task, but is filled in for its Server task.

The Cumulative Statistics shown in this example are exactly the same as those shown in the previous example for the Server task. This is because the counts for connections and data transmissions are accumulated based on the local port number of the Servers. The Listener task and all of its subordinate Server tasks (which it started) all have the same local port number, and will all show the same set of Cumulative Statistics.

Client UDP Transaction Detail

This third example is a UDP Client transaction that sends a datagram to the echo port of a remote host.

This figure shows a page of the **Browse Detail** panels invoked by the **Browse Clients** function:

1	04/27/94 18:39:48	2	CLIENT/SERVER BROWSE DETAIL	BD01	3
		4	CLIENT TRANSID U0S2 TO PORT 7	P. 1	
		5	STATISTICS CAPTURE CURRENTLY OFF		
6	TASK#: 35 LPORT: 4365 TOKEN: 00098528 PROTOCOL: UDP STATE: RCFR---OK				
	FIRST ACTIVITY@: 04/27/94 14:08:21 LAST ACTIVITY@: 04/27/94 18:35:56				
	REMOTE ADDRESS: 138.70.45.12:7				
	HOST NAME: BART.INTERLINK.COM.				
	MESSAGES RECV'D: 1		SENT: 1		
	BYTES RECV'D: 15		SENT: 15		
	(MORE) 7				
8	>				
	Pf1=? Pf3=goback Pf4=Utilization Pf5=Configuration Pf8=Fwd				
9	Pf2=End Task (close) Pf6=Ping Remote Pf9=Trace Options				

ITEM	DESCRIPTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID and current page number.
4	The transaction ID of the Client task and the local port it is using.
5	The cumulative statistics have been turned off at CPT initialization.
6	The identification line for this UDP endpoint, giving its task number, local port number, CEP token address, protocol, and current state.
7	The command line used for specifying an Administrator Interface function code.
8	The (MORE) indicator that more panels are available as part of a browse.
9	The PF keys currently active for this panel.

Displaying a UDP Endpoint

A UDP task has an endpoint, but no connection with a remote host, whether it is a Client or a Server transaction. Because a UDP client must specify a remote host and port number, that information can be displayed, along with the endpoint statistics for amount of data sent and received by the UDP task.



Note: The top half of the screen is empty where the Cumulative Statistics would be shown. This is not because the Client is a UDP task, but because `AISTATOP=NO` has been specified in the CPT Configuration Table for this execution of CICS. When this parameter is set to `NO`, every **Browse Detail** panel will have this message displayed:

STATISTICS CAPTURE CURRENTLY OFF.

Terminating the Endpoint Task

The **PF2** key is used to terminate the CICS task for the endpoint displayed on the **Browse Detail** panel. This termination is done by an abortive close on the CPT endpoint, not by abending the CICS task explicitly. This action will cause a disconnect return code to any outstanding API call from the CICS task, which must then interpret that disconnect return code, and terminate via its own logic. When the **PF2** key is hit, the **Quiesce/Confirm** panel is displayed to confirm the close of the displayed task.

04/27/94 18:40:59		QUIESCE/CLOSE CONFIRMATION			ETO1
TASKNO	PORT	REMOTE HOST:PORT	TOKEN	TRANID	
212	4394	138.70.45.12:0007	000C59B8	TPC2	
PLEASE USE PF9 (QV) TO CONFIRM REQUEST TO QUIESCE THIS TASK!					
> Pf1=? Pf3=goback Pf4=Utilization Pf5=Configuration					

Confirming the Quiesce

The **PF9** key confirms the termination, or the `QV` function code can be entered on the command line. After confirmation, the next panel displayed is the **Utilization Summary** panel, to verify the current status of the task. This response message displays in the message area of that panel:

>REQUESTED CONNECTION(S) CLOSED

If the termination is to be aborted by not confirming on the **Quiesce/Confirm** panel, any of the standard navigation function codes or PF keys can be used to go to another panel. The **PF3** key returns to the **Utilization Summary** panel, not the previously displayed **Browse Detail** panel.

Pinging the Remote Host

Use the **PF6** key to send a ping to the Remote Host displayed on the **Browse Detail** panel. The ping is an ICMP echo request that does not use the displayed connection, but can test if the Remote Host is still reachable and if its system

20124030-200100

is still active. If a break has occurred in the route to the host, or if its system has crashed, this message displays on the Browse Detail panel:

```
>          ....NO RESPONSE FROM REMOTE HOST....
```

If the ping is successful, then this message displays:

```
>          ....REMOTE HOST IS ALIVE....
```

This feature can help in determining if the endpoint being displayed needs to be terminated because the Remote Host is no longer reachable.

Setting Trace Options

Probably the most powerful action available on the **Browse Detail** panel is the ability to dynamically set Trace Options for the endpoint currently being displayed. Use the **PF9** key to go to the **Online Trace Options** panel.

These trace options are the same API trace options that can be set in the ACM control block when an API program first issues a `LISTEN` or `CONNECT` call. These trace options can be set and removed dynamically without recompilation of the program.

When any API trace option is set, the trace records associated with that API event will be written to the ACTR Transient Data queue for every transaction executed. These records will continue to be written as long as the trace options are set in the **Online Trace Options** panel, described in the following section.

Online Tracing

The new Online Tracing facility allows dynamic setting and removal of trace options, and the online browsing of the trace output from within the Administrator Interface. The options for tracing are the same parameters specified in the Configuration Table for CPT tools, or in the ACM control block for API programs. The output records of these logging traces are sent to a Transient Data queue specified by the `T09MCICS` macro in the Configuration Table, whose default name is ACTR.

With the Online Trace, the same records that are sent to the ACTR log, can also be browsed online. These Online Trace records are kept in a Temp Storage queue that can be browsed from the **Online Trace** panel. This allows for a more dynamic diagnosis of a TCP connection, without requiring the recompilation of an API program, or the re-assembly of the Configuration Table to turn the trace options on and off.

The Online Tracing options can be set and removed for a particular endpoint at these places within the Administrator Interface:

- ◆ **Online Tracing Options** panel
PF9 off the **Browse Detail** panel
- ◆ **Generate Trace Entry** panel
PF6 off the **Utilization Summary** panel

The first is for tracing endpoints that are currently established in the system, and the second is for tracing an endpoint that does not have an active transaction. Both of these methods are discussed in detail in the following sections.

Controlling the Size of the Online Trace Queue

In order to collect the Online Trace entries, a parameter must be specified in the Configuration Table. This parameter of the `T09MCICS` macro is described further in **Appendix A – CONFIGURATION STATEMENTS**.

- ◆ `AITSTRAC=nnn`
Specifies the maximum number of trace records that can be written to the Temp Storage queue for Online Trace. The default is zero, with Online Trace turned off.

The size of this queue for trace records can be dynamically changed from the **Operations Menu** of the Administrator Interface. The maximum size allowed is 999 records.

Online Trace Options Panel

The **Online Trace Options** panel allows API trace options to be dynamically set and removed for a specific CPT endpoint. It is also where the Online Trace for that specific endpoint is acquired, if the Online Trace is not currently being used to trace another endpoint.

1
04/27/94
18:42:30

2
ONLINE TRACE OPTIONS

3
TO01

4
TRAN: TPC1 PORT: 1804

API STATS==>

5
CONN: Y TERM: Y

API TRACE==>

6
ENTR: Y ARGS: _ RECV: Y SEND: _ TERM: _ PASS: _
TOKN: _ STOR: _ TPL: _ TERR: _ CLOS: Y CLTD: _
RLSE: Y

7 ONLINE TRACE==> Y

8
** ONLINE TRACE NOW ACQUIRED FOR THIS CONNECTION **

9
Pf1=? Pf2=refresh Pf3=goback (nosave)
Pf9=confirm/goback Pf12=confirm/watch trace

ITEM	DESCRIPTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID.
4	The transaction ID of the Server task and the local port it is using.
5	The statistics trace option settings for this endpoint.
6	The API trace option settings for this endpoint.
7	The command prompt for acquiring and releasing the Online Trace.
8	The message area for command feedback and error notification.
9	The PF keys currently active for this panel.

Trace Option Specifics

The trace options are exactly the same as the trace options specified for CPT tools in the Configuration Table and for API programs in the ACM control block. The advantage here is that they can be dynamically set and removed. Setting these trace options will cause trace records to be written to the trace log queue, ACTR.

Statistics Trace Options.

The statistics options generate trace entries at the creation and termination of a Connection endpoint.

- ◆ **CONN** – This option will generate a statistics message when a new endpoint is established for a client or server task.
- ◆ **TERM** – This option will generate a statistics message when an endpoint is terminated for a client or server task.

API Trace Options.

The API options generate trace entries whenever specific events occur for an endpoint within the CPT Task Related User (TRU) Exit routines.

- ◆ **ENTR** – This option will generate a message upon entry to each TRU Exit routine.
- ◆ **ARGS** – This option will generate a hex dump of the caller's argument upon entry and exit from a TRU Exit routine.
- ◆ **RECV** – This option will generate a hex dump of the transport provider's input buffer that is received.
- ◆ **SEND** – This option will generate a hex dump of the transport provider's output buffer that is sent.
- ◆ **TERM** – This option will generate a message upon exit from each TRU Exit routine.
- ◆ **PASS** – This option will generate a hex dump of all resources passed for the **GIVE** and **TAKE** calls.
- ◆ **TOKN** – This option will generate a hex dump of the CEP control block upon entry to each TRU Exit routine.
- ◆ **STOR** – This option will generate a hex dump of the argument used upon entry and exit from the TRU **EXIT** routine for storage management.
- ◆ **TPL** – This option will generate a hex dump of the TPL control block for each call to the transport provider.
- ◆ **TERR** – This option will generate a hex dump of each TPL control block after an error is returned from a transport provider call.
- ◆ **CLOS** – This option will generate a hex dump of all resources related to a closed endpoint.

- ◆ **CLTD** – This option generates a hex dump of the Transient Data record, written by a Client-Data Listener when it receives input data in the Transient Data format.
- ◆ **RLSE** – This option will generate a message when a TRU Exit routine receives a release indication from the transport provider.

Acquiring and Releasing the Online Trace

The Online Trace can only be used to trace one connection endpoint at a time. And the trace can only be observed from one user terminal at a time. The Online Trace facility must be released by one user terminal before it can be acquired and watched from another user terminal.

The Online Trace is acquired by entering a 'Y' at the command prompt, as shown here:

```
ONLINE TRACE==> Y
```

```
** ONLINE TRACE NOW ACQUIRED FOR THIS CONNECTION **
```

If the Online Trace has already been acquired, the command prompt will not be shown. Instead, an information message giving the local port number of the traced connection, and the VTAM name of the terminal currently using the Online Trace are displayed, as shown here:

```
ONLINE TRACE OF 1804 AT L0910702 ACTIVE
```

The Online Trace must be released by the original terminal user before it can be acquired for another endpoint. This is done by entering an 'N' in the command field, as shown here:

```
ONLINE TRACE==> N
```

```
** ONLINE TRACE NOW RELEASED FROM THIS CONNECTION **
```

Configuration Table Initialization Parameter

If the **AITSTRAC** parameter has not been specified in the Configuration Table, or if the number of trace entries has been set to zero from the **Reset Trace** panel, then the Online Trace cannot be used. The command prompt will not be shown, and the following message will be displayed:

```
ONLINE TRACE NOT AVAILABLE
```

The Online Trace can easily be made available again by using the **Reset Trace** panel off the **Operations Control** menu, as described in **Reset Online Trace Queue Function** on page 4-59.

The Online Trace Panel

After the Online Trace has been acquired for tracing an endpoint, the **Online Trace** panel can be entered by using **PF12** from the **Online Trace Options** panel. The **Online Trace** panel displays all the trace records generated for the options selected for that endpoint.

This figure is an example of one page of a **Online Trace** panel browse:

① 04/29/94 14:47:08	② ONLINEe TRACE TRAN: TPC1 PORT: 1804	③ WT01
⑤ 00058 0000 000EF7B8 0028		
*129) CPT901I MODULE T09CRECV 04/12/94 09.40 A1011449 ENTERED AT 14:45:52 94.11		
⑥ *130) CPT913I T09CRECV HEX DUMP OF API TREC V TPL DATA		
*131) 0000 000F0800 20202020 20202020 20202020 20202020 20202020 20202020 2020		
*132) 0020 000F0820 20202020 20202020		
*133) CPT901I MODULE T09CXLAT 04/12/94 09.47 A1011449 ENTERED AT 14:45:52 94.11		
*134) CPT901I MODULE T09CSEND 04/12/94 09.42 A1011449 ENTERED AT 14:46:45 94.11		
*135) CPT901I MODULE T09CXLAT 04/12/94 09.47 A1011449 ENTERED AT 14:46:45 94.11		
*136) CPT901I MODULE T09CSEND 04/12/94 09.42 A1011449 ENTERED AT 14:46:45 94.11		
*137) CPT901I MODULE T09CRECV 04/12/94 09.40 A1011449 ENTERED AT 14:46:45 94.11		
*13⑦) CPT913I T09CRECV HEX DUMP OF API TREC V TPL DATA		
*139) 0000 000F1848 0028		
*140) CPT913I T09CRECV HEX DUMP OF API TREC V TPL DATA		
*14⑧) 0000 000EE770 20202020 20202020 20202020 20202020 20202020 20202020 2020		
*142) 002⑨) 000EE790 20202020 20202020		
*143) CPT901I MODULE T09CXLAT 04/12/94 09.47 A1011449 ENTERED AT 14:46:45 94.11		
⑩ > Pf1=?	⑪	Pf7=Bwd Pf8=Fwd
⑫ Pf2=Show First	Pf6=Window Left/right	Pf9=Show Last Pf12=Watch Release

ITEM	ACTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID.
4	The transaction ID of the Server task and the local port it is using.
5	The CICS task number of the endpoint task generating the traces messages.
6	A line count of subsequent messages produced by the same task.
7	The CPT message number for each new trace message.
8	The hexadecimal displacement field for each line of a hex dump message.
9	The hexadecimal address field for each line of a hex dump message.
10	The command line used for specifying an function code.
11	The message area for command feedback and error notification.
12	The PF keys currently active for this panel.

20
24030-200100

Browsing the Online Trace

When the Online Trace queue reaches the end of its available entries, the trace messages will wrap around and begin overlaying the first messages in the queue. The line numbers for the messages will also wrap when they reach 999, which is the maximum allowable entries in the Online Trace.

The PF key selections allow the user to browse forward and backward through the messages, to scroll the display left and right, and to jump to the first or last message in the queue. When finished with browsing the trace messages, the **PF12** key can be used to exit the **Online Trace** panel.

Releasing the Online Trace

The option, **PF12=Watch Release**, does not release the Online Trace. It merely releases the watching by exiting the **Online Trace** panel and returning to the **Utilization Summary** as its next panel. In order to release the Online Trace, you must reselect the task that was being traced, press **PF9** to display the **Online Trace Options** panel for that endpoint, and then specify:

```
ONLINE TRACE==> N
```

This action will then release the Online Trace so that it can be used to trace another endpoint, or be used from another terminal using the Administrator Interface.

The Generate Trace Entry Panel

The **Generate Trace Entry** panel sets and releases API trace options for a endpoint that may not be currently displayed on the **Utilization Summary** panel because the client or server task has yet not begun its execution.

This facility allows for tracing connections that are short-lived and cannot be paused in their execution. The endpoint cannot be identified by its **TASKNO** or **TOKEN** since the task does not exist yet, so the endpoint must be identified by a combination of its CICS transaction ID, the local or remote port number, and the protocol used when the endpoint is established. The illustration below is an example of the **Generate Trace Entry** panel.

1 04/27/94
18:50:46

2 GENERATE TRACE ENTRY

GT0 **3**

TO ESTABLISH AN ENTRY FOR SETTING TRACE OPTIONS

 PLEASE ENTER THE CLIENT OR SERVER

4 WITH FORMAT 'TTTT,XNNNNNP'

 TTTT = TRANSACTION NAME <MUST BE 4 BYTES>

 X = L(OCAL)-OR-R(EMOTE) NNNNN = PORT NUMBER

 P = T(CP)-OR-U(DP) PROTOCOL <DEFAULT=TCP>

5 > TP91,L1004T
7 Pf1=? Pf3=goback

6
 Pf4=Utilization Pf5=Configuration

ITEM	ACTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID.
4	The instructions for formatting the trace entry specification.
5	The command used for entering the trace entry specification.
6	The message area for command feedback and error notification.
7	The PF keys currently active for this panel.

Entering the Trace Parameters

As the panel instructions indicate, you must enter a string of two parameters on the command line. If the specification is valid, the **Online Trace Options**

panel will be displayed. If the specification is invalid, one of the following error messages is displayed.

The transaction ID is edited to make sure that it is four characters. A CICS lookup is done to make sure it is a valid transaction ID. If it is found invalid, this message appears:

```
TRANSACTION NOT DEFINED TO CICS: HUMF,L2002
```

Client transactions must specify the remote port number to which they will connect on the remote host, and Server transactions must specify the local port number on which they will listen or receive. If the first character of the second parameter is not L or R, this message displays:

```
PORT MUST BE DESIGNATED LOCAL OR REMOTE: IPTL,2002
```

The port number must be 5 digits or less (no leading zeros are required) and must be a number between 1 and 65,534, or this message displays:

```
PORT MUST BE 1-65534: IPTL,L99999
```

The second parameter's suffix for the trace entry specifies the protocol, TCP or UDP. TCP is the default if no suffix is specified. If a character other than T or U follows the port number, this message displays:

```
REQUESTED PROTOCOL MUST BE TCP OR UDP: IPTL,L2002X
```

Selecting Trace Options

After the trace specification is accepted, the **Online Trace Options** panel will be displayed to set the trace options, as described in **Trace Option Specifics** on page 4-22. Setting these API trace options will cause trace records to be written to the ACTR trace log for every instance of this transaction and port until the trace options are removed.

After setting the options, the Online Trace can be acquired, if it is available, by entering 'Y' at the command prompt, as shown here:

```
ONLINE TRACE==> Y
```

```
*** ONLINE TRACE NOW ACQUIRED FOR THIS CONNECTION ***
```

Releasing the Generated Trace Entry

Once the trace options have been specified on the **Online Trace Options** panel and the Online Trace has been acquired for that entry, the **Utilization Summary** panel displays again. There is now an inactive endpoint displayed on that panel for the transaction ID, port number, and protocol just specified, as shown in this example:

TOKEN	TASKNO	TRANID	PORT	REMOTE HOST:PORT	STATE
000251B0		TP91	1004 TCP		INACTIVE

In order to select this endpoint now, to change the trace options, watch the trace, or release the trace, the address in the **TOKEN** field can be specified to get the **Browse Detail** panel for this inactive endpoint. The **Online Trace Options** panel can then be selected with **PF9**, as was done for active endpoints.

Reset Trace Online Queue Panel

If you cannot remember which endpoint currently holds the Online Trace, or if you cannot access the terminal from which it was acquired, you can still release the Online Trace with the **Reset Trace** panel. You can display this panel with the **PF11** key off the **Operations Control** menu, or you can jump to it directly by entering the **RT** function code on any panel command line. The illustration below is an example of the **Reset Trace** panel.

1	04/27/94 19:15:48	2	RESET TRACE/ERROR ONLINE QUEUE	RQ01	3
4 ENTER NUMBER 0,30-999 TO CHANGE SIZE OF T/S QUEUE FOR TRACE					
AND/OR					
5 USE PF9 TO CONFIRM REINITIALIZATION OF THE QUEUE					
6	> 999				
7	Pf1=? Pf3=goback Pf4=Utilization Pf5=Configuration				

ITEM	ACTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID.
4	The instructions for setting the size of the Online Trace queue.
5	The PF9 Confirmation request to reinitialize the queue.
6	The message area for command feedback and error notification.
7	The PF keys currently active for this panel.

Confirming the Reset Trace

If you merely enter **PF9** on this screen, you purge the Online Trace queue of its accumulated trace messages and release the queue if it had been acquired for a particular endpoint. This confirmation returns you to the **Utilization Summary** panel.

If you specify a new queue size on the command line and press **Enter**, you return to the **Utilization Summary** panel, with this message shown:

TRACE QUEUE SIZE WILL CHANGE AT NEXT INITIALIZATION

If you specify a new queue size on the command line and press **PF9**, you return to the **Utilization Summary** panel, with this message shown:

TRACE ONLINE QUEUE HAS BEEN RESET AND SIZE CHANGED



Note: You cannot specify an online trace queue size less than 30 or greater than 999. And, if you set the size to zero, Online Trace is turned off.

Configuration

The Configuration component of the Administrator Interface provides an ability to display and modify the CPT tool definitions made in the CPT configuration table. In the previous version of CPT, these tool definitions could only be changed by reassembling the configuration table macros. The Administrator Interface now lets you modify the definitions of the CPT tools in these ways:

- ◆ Update the parameters of any Listen/Receive tool
- ◆ Add a new Listen/Receive tool to the configuration table
- ◆ Start any of the Listen/Receive tools defined to the configuration table
- ◆ Update the parameters of any Send tool
- ◆ Add a new Send tool to the configuration table

Some global parameters of the configuration table are also displayed by the Configuration component, but they cannot be modified. This is because they define the attributes of the currently running CPT Interface (such as the subsystem ID of its TCP transport provider), which cannot be modified without stopping and restarting the CPT Interface.



Note: The dynamic updates to the tool parameters, and the addition of new tool definitions, are not permanent changes to the Configuration Table. The next time the CPT Interface is initialized, the Configuration tools will revert to those definitions assembled into load module T09CONFIG.

Configuration Table Parameters Panel

The **Configuration Table Parameters** panel displays all of the global attributes of the CPT Interface that is currently running. These attributes are specified by the T09MCICS macro coded at the beginning of the configuration table. This panel shows both the default attributes, and those that have been set by macro parameters for the current definition of the CPT Interface. These parameters cannot be modified dynamically.

The **Configuration Table Parameters** panel is invoked by the **C** function code from anywhere in the Administrator Interface, or by using the **PF5** key, when this option is displayed at the bottom of a panel:

PF5=Configuration

This is an example of the **Configuration Table Parameters** panel:

1	04/27/94	2	CONFIGURATION TABLE PARAMETERS	CT01	3
	18:54:32	4	CPT INTERFACE STARTED AT 14:04 ON 04/27/94 IN APPLID CICSQA01		
5	LOCAL HOST NAME: HOMER.INTERLINK.COM.				
	ADDRESS: 138.99.128.13				
6	TCP/IP SUBSYSTEM: ACSS		7	DNR SUBSYSTEM: ACSS	
	CICS AMXT: 100				
	TRANSACTION NAMES: TERMINATION = IPPR LISTEN = IPTL				
9	TD QUEUE NAMES: STATISTICS = ACST TRACE = ACTR ERRORS = ACER				
10	TRANSLATION TABLE: T09XENG				
	ADM INT STATISTICS TABLE: 8,000 BYTES 165 ENTRIES				
	STATISTICS DISCARD TDQ: ACOS				
11	MAX TRACE RECORDS: 800				
	MAX ERROR RECORDS: 500				
	COMPONENT NAME PREFIX: IPA				
	OPTIONS: CAPTURE CLIENT/SERVER STATISTICS				
12	>				
14	Pf1=?	Pf3=goback	Pf4=Utilization	Pf5=Configuration	
			Pf2=Listen Tools	Pf6=Send Tools	

ITEM	DESCRIPTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID.
4	The date and time that the CPT Interface was initialized, and the APPLID of this CICS system.
5	The local host name and IP address of the TCP transport provider, to which this CPT Interface is connected.
6	The subsystem ID by which the TCP transport provider is specified.
7	The subsystem ID of the Domain Name Resolver for IP address/name translations.
8	The CICS AMXT parameter from the SIT table, specifying Active Max Tasks.
9	The transaction names and transient data queue names used by CPT.
10	The load module containing the ASCII/EBCDIC translation table.
11	The parameters for the Administrator Interface itself.
12	The command line used for entering a function code.
13	The message area for command feedback and error notification.
14	The PF keys currently active for this panel.

Configuration Table Description

Most of these attributes will rarely change for the CPT Interface, and this list provides a quick overview of them before browsing the CPT tool definitions. A brief description of each field is also given here.

Local Host Name

The full domain name of the local host's TCP/IP subsystem.

Address

The dotted decimal IP address of the local host's TCP/IP subsystem.

TCP/IP Subsystem

The subsystem ID of the TCP/IP transport provider.



Note: The local host name and address are not explicitly specified in the configuration table. They are derived from the `SSN=` parameter that specifies which subsystem is going to provide TCP/IP services for this CICS.

DNR Subsystem

The subsystem ID of the Domain Name Resolver for IP Name / IP Address translation.

CICS AMXT

The CICS Active Max Task value from the CICS system initialization table.



Note: This CICS parameter is shown here because Listener tasks are long running tasks. Creating and starting new Listener tools could adversely impact a CICS system with a low AMXT value.

Transaction Names

The configuration table lets you change these transids, if the default names are already used by this CICS.

2012-03-24 03:00:00

TD Queue Names

The configuration table lets you change these queue names, if the default names are already in use by this CICS.

Translation Table

The configuration table lets you specify a custom ASCII/EBCDIC translation table, if the default table provided is not sufficient.



Note: The **Configuration Table Parameters** panel illustration shows the default values for all the transaction names, queue names, and translation table.

Adm Int Statistics Table

This field shows the number of bytes and the number of entries allocated to the internal statistics table used by the Administrator Interface. The default is 1000 bytes.



Note: The **Configuration Table Parameters** panel illustration shows the values when parameter `AITABSZ=8000` is specified in the configuration table.

Statistics Discard Queue

When the Cumulative Statistics are reset, or if CPT is shut down, the entries in the statistics table are written as data records to this transient data queue. The default is no queue defined.



Note: If the queue name specified in this field is not defined in the CICS DCT, then this error message appears:

TRANSIENT DATA QUEUE XXXX IS NOT DEFINED TO CICS

Max Trace Records

This field displays the maximum trace entries that can be collected by the Online Trace. This value can be reset from the **Operations Control** menu. The default is 0.

Max Error Records

This field displays the maximum Error Log record entries that can be displayed from the **Operations Control** menu. The default is 30.

Component Name Prefix

This field displays the 3-character prefix of all the transids used for the Administrator Interface. IPA is the default prefix.

Options

This field is displayed when Cumulative Statistics are being collected for CPT transactions.



Note: The **Configuration Table Parameters** panel illustration shows the option line displayed. If `AITSTATOP=YES` is not specified in the configuration table, this line does not display.

At the bottom of this panel are the PF key selections for browsing the CPT tools defined by this configuration table. **PF2** begins the browse of the Listen/Receive tools, and **PF6** begins the browse of the Send tools.

Listen/Receive Tool Panel

In the batch definition of the configuration table, the Listen and Receive parameters are defined by separate macros. But here, those parameters are combined on a single panel, since they must be defined together as a single server tool to receive incoming data from remote clients. From this panel, four potential actions can be taken on this tool:

- ◆ Start the Listener task for the tool, if it had been quiesced before.
- ◆ Alter the parameters defining the Listen/Receive tool.
- ◆ Add a new Listen/Receive tool, based on a current tool definition.
- ◆ Set the Statistics and Trace options dynamically for this tool definition.

Examples of these actions are given after an explanation of the fields of the **Listen/Receive Tool** panel shown in this example:

```

1 04/27/94                2 LISTEN/RECEIVE TOOL                LT01
18:57:23 4 CPT INTERFACE STARTED AT 14:04 ON 04/27/94 IN APPLID CICSQA01 P. 1
5 LOCAL HOST NAME: HOMER.INTERLINK.COM.
  ADDRESS: 138.99.128.13
6 LISTEN ON WELL-KNOWN PORT: 3380
  BUFFERS: 4 @ 1024 BYTES
  DNR CALLS: YES      SYNCPOINT: NO      CLIENT TIME:

RECEIVE TRANSACTION NAME: IPTR      PROGRAM NAME: T09TREC
7 RECEIVE TD QUEUE: XXRF      TD QUEUE BUFFER: 1024 BYTES
  RECEIVE OPTIONS: FILE,RAW

8 TRAN STATS: NONE
  TRAN TRACE: NONE

  API STATS: CONN,TERM
  API TRACE: ARG,ENTER,TERM,SEND,RCV,PASS,CLOSE,TPLERR,TOKEN,TPL,STOR

10 WARNING: TRANSIENT DATA QUEUE XXRF IS NOT DEFINED TO CICS
9 >
  Pf1=?   Pf3=goback   Pf4=Utilization   Pf5=Configuration   Pf8=Fwd
11 Pf2=Tool Trace Pf6=Send Tools   Pf9=Add/alter Listen   Pf10=Listen Start
  
```

ITEM	DESCRIPTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID and page number.
4	The date and time that the CPT Interface was initialized, and the APPLID of this CICS system.
5	The local host name and IP address of the TCP transport provider, to which this CPT Interface is connected.
6	The parameter fields derived from the T09MLSTN macro definition.

ITEM	DESCRIPTION
7	The parameter fields derived from the T09MRECV macro definition.
8	The transaction and API trace options for this tool.
9	The command line used for entering a function code.
10	The message area for command feedback and error notification.
11	The PF keys currently active for this panel.

Many of the tool panels appear quite similar at first glance. The key fields to focus on here are the port number, which defines where the TCP connection will be made, and the Receive TD queue, which defines where the incoming data will be written. The fields shown on this panel are described next.

Local Host Name

The full domain name of the local host's TCP/IP subsystem.

Address

The dotted decimal IP address of the local host's TCP/IP subsystem.



Note: The local host name and address are not specified in the macro definition for the Listen/Receive tool. These fields are being replicated here from the initial **Configuration Table Parameters** panel for reference purposes only.

Well-Known Port

This port number is where the Listener task accepts new connections from remote clients.



Note: In the macro definition, this port number can also be specified by its service name, which is translated by DNR into this port number.

Buffers

This field defines the number and size of receive buffers to be allocated for input data from the connection. Four buffers at 1024 bytes each is the default.

DNR Calls

This field shows if the Listener task will automatically issue a DNR call, when a new connection is established, to translate the remote client's IP address into its domain name.

Syncpoint

This field shows if the Listener task will issue a CICS Syncpoint call prior to starting the Receive Tool or transaction.

Client Time

If a timeout value is specified by this field, it indicates that the Client-Data Listener option will be used for this Listen Tool. This option implies that the Listener will do a timed-receive on the input datastream to determine what transaction should be started by the Listener. Detailed information about this option is described in **Appendix A – CONFIGURATION STATEMENTS**.



Note: When the Client Time parameter is specified, none of the Receive Tool parameters are displayed. This is because there can be no associated Receive Tool or transaction ID specified for this type of Listen Tool in the `T09MCICS` macro. The Client-Data Listener task will receive its transaction ID to start from the input datastream, similar to the 3270 transaction input to CICS.

Receive Transaction

When the Listener gets a new connection, it starts this transaction to receive data on the connection.

Receive Program

This field is not macro-specified, but the Receive program can be determined based on the Receive Transaction. It is displayed here for reference purposes. Program `T09TREC` is the default Receive Tool program.

Receive TD Queue

This is the Transient Data queue to which incoming data is written as TD records.

TD Queue Buffer

This is the maximum record size that can be written to the TD queue. The default is 1024 bytes.

Receive Options

These options define how the TD records will be separated from the input data stream. These options are described in the `T09MRECV` macro definition in **Appendix A – CONFIGURATION STATEMENTS**.

Tran Stats & Trace

These options define what transaction-level tracing and statistics are should be written for this tool. The options are described in the `T09MLSTN` macro parameters, `TRNSTAT` and `TRNTRAC`, in **Appendix A – CONFIGURATION STATEMENTS**.

API Stats & Trace

These options define what API-level tracing and statistics are should be written for this tool. The options are described in the `T09MLSTN` macro parameters, `APISTAT` and `APITRAC`, in **Appendix A – CONFIGURATION STATEMENTS**.



Note: The statistics options specify which stats records are written to the TD queue for statistics, whose default queue name is ACST. The trace options specify which trace records will be written to the TD queue for trace logging, whose default queue name is ACTR. The trace records will not be written to the Online Trace, unless a specific trace request is made through that facility.

Warning Messages

This panel will check to see if the Receive Transaction and TD queue specified are defined to CICS. If not, a warning message is displayed.



Note: The **Listen/Receive Tool** panel illustration includes a warning line, because the Receive TD queue was defined in the `T09MRECV` macro, but was not defined in the CICS Destination Control Table.

The parameters displayed for this tool can be modified by hitting the **PF9** key. This will take you to the **Add/Alter Listen Tool** panel described below. That same panel is used to modify the current tool definition or to create a new tool definition, which is why **PF9** has this label at the bottom of the panel:

Pf9=Add/alter Listen

Starting the Listen Tool

All the Listen/Receive tools defined in the configuration table are started at CPT initialization time. However, if any of those Listener tasks was stopped from the **Browse Detail** panel, or quiesced from the **Operations Control** menu, it is difficult to get them started again as a long-running task that is not associated with any terminal.

The **PF10** key on the **Listen/Receive Tool** panel provides the ability to start the Listen/Receive tool currently displayed. After **PF10** is pressed, the following message is displayed:

```
...LISTENER HAS BEEN STARTED...
```

As with any Listener task, if there is already a task listening on the port number specified for this tool, the started task will fail and write these error messages to the Error Logging TD queue, ACER:

```
CPT236E (TBIND FAILED...)
```

```
CPT226E (CPT LISTEN ROUTINE FAILED...)
```

Add/Alter Listen Tool Panel

This panel provides the ability to modify the parameters of the currently displayed Listen/Receive tool, or the ability to create a new Listen/Receive tool definition based on the current display. The key field that determines if this is an Add or an Alter of the tool is the port number field. The initial message displayed on entry to this panel tries to explain this:

CPT LISTEN/RECEIVE TOOL WILL BE ADDED IF NEW PORT IS ENTERED

1	04/27/94 18:59:16	2	ADD/ALTER LISTEN TOOL	3	TA01
4	WELL-KNOWN PORT: 3380	OR SERVICE:			
5	RECEIVE BUFFERS: 4 @ 1024 BYTES DNR CALLS: SYNCPOINT: YES	CLIENT TIME:			
6	TRANSACTION: IPTR				
RECEIVE TOOL PARAMETERS					
7	TRANSIENT DATA QUEUE: IPRF		BUFFER SIZE: 1024 BYTES		
8	TRANSLATE TABLE: DEFAULT				
9	..PROCESSING OPTION: SEP=CRLF				
10	STATS OPT==> TRM: Y TRACE OPT==> ENT: Y ARG: _ RTD: _ WTD: _ TRM: Y TKN: _ XLA: _				
11	PARAMETERS ARE VALIDATED - USE PF9 TO CONFIRM ADD/ALTER				
12	Pf1=? Pf3=goback (nosave) Pf2=refresh Pf9=tool save				

ITEM	DESCRIPTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID.
4	The local port number that identifies this Listen/Receive tool.
5	Receive buffer number and size.
6	The Receive transaction ID is IPTR.
7	The Receive Transient Data queue name and its maximum record size.
8	The Translate Table parameter is used to specify a unique ASCII/EBCDIC translation table for this tool.
9	The Processing Options field is used to specify how the TD records are created from the input datastream.
10	Stats and Trace options are set on and off dynamically for this Listen/Receive tool.
11	The message area for command feedback and error notification.

ITEM	DESCRIPTION
12	The PF keys currently active for this panel.

**Add/Alter Listen Tool
Field Descriptions**

Since the Listener task is uniquely defined by the port it listens on, by changing the port number on this screen, you are implying that you want to add a new tool definition that will use that port number. Of course, you will probably also want to change the Transient Data queue name to which the Receiver task will write its incoming records, and perhaps also the record separation parameters defined for that queue.

Port Number

The port number for a new tool can be explicitly entered, or a Service name can be entered, which will be translated to a port number by the Domain Name Resolver. The port number will be edited, and must be a number between 1 and 65,534. Otherwise, the following error message appears:

== PORT NUMBER MUST BE BETWEEN 1 AND 65534==

Receive Buffers

The Receive Buffer number and size are edited for numerics, and have a total buffer size limit of 61K, or the following error message appears:

== BUFFERS * SIZE MUST NOT BE MORE THAN 61K ==

DNR Calls

Specify **no** in this field to suppress the automatic DNR call made by the Listener task for each new connection. **yes** is the default value. Any other value will display the following error message:

== DNR CALL OPTION MUST BE 'YES' OR 'NO' ==

Syncpoint

Specify **yes** in this field to require a CICS syncpoint call prior to starting the Receive transaction. **no** is the default. Any other entry will display the following error message:

== SYNCPOINT OPTION MUST BE 'YES' OR 'NO' ==

Client Time

Specify the number of seconds for the timeout value of a Client-Data Listener. If the field is blank and a timeout value is entered, this implies that you want to

add a new Listen Tool with the Client-Data option, just as if you entered a new port number. The default value is zero, which means there is no Client-Data option for this Listen Tool.

Receive Transaction

The default Receive transaction ID is IPTR, but if another transaction is specified it must be defined to CICS via RDO or the PCT, or the following error message appears:

```
== WARNING: TRANSACTION XXXX IS NOT DEFINED TO CICS
```

Receive Transient Data Queue

The Transient Data queue name is probably the most important field to set besides the port number. However, if you are adding a new tool, you must make sure that the TD queue name has already been defined in the CICS Destination Control Table, or you will get the following error message:

```
WARNING: TRANSIENT DATA QUEUE YYYY IS NOT DEFINED TO CICS
```



Note: There is no restriction on more than one tool writing into the same TD queue, but you should make sure that the CICS program reading that queue can distinguish the different types of records written into it.

Buffer Size

The buffer size of the Transient Data queue defines the maximum record size that can be written to the specified queue. If processing option, `FILE`, is specified, that means the entire input datastream for the connection must fit into one record. So this buffer must as large as the largest data transmission expected. This field is edited for numerics, and the CICS maximum record size for transient data records is 32,767 bytes.

Translate Table

The Translate Table parameter is used to specify a unique ASCII/EBCDIC translation table for this tool. Otherwise, the default table specified in the `T09MCICS` macro is used. There is no editing done on this panel field. The two common entries for this field are:

- ◆ `RAW` – No translation is done on the input data stream
- ◆ `DEFAULT` – The default translation module, `T09XENG`, will be used.

Processing Options

The Processing Options field is used to specify how the TD records will be created from the input datastream. One of the four basic options defined by the `OPTIONS` parameter of the `T09MRECV` macro should be specified:

- ◆ **LL** – A length field will be used to determine record boundaries.
- ◆ **SEP** – A separator character will determine record boundaries.
- ◆ **FILE** – The whole datastream will be written as a single record.
- ◆ **ALL** – The whole datastream will be written as a single record.
- ◆ **STRIP** – The separator character will not be written with the record (this is the default for **SEP**).
- ◆ **NOSTRIP** – The separator character will be written with the Transient Data record.



Note: If option **SEP** is specified, the specific **SEP** characters are defined following an equal sign, just as the **SEP=** parameter is specified in the **T09MRECV** macro. The default specification is **SEP=CRLF**, which means the incoming records are separated by ASCII carriage return and line feed characters, which are hex '0D0A'.

See the **T09MRECV** macro in **Appendix A – CONFIGURATION STATEMENTS** for other valid **SEP=** specifications. If an invalid parameter option is specified in this field, you will also be referred to that macro definition in **Appendix A – CONFIGURATION STATEMENTS** by this error message:

```
== ERROR DETECTED IN OPTIONS - SEE MANUAL FOR SPECIFICATIONS
```

Stats and Trace Options

The Stats and Trace Options are set on and off dynamically for this Listen/Receive tool by entering either a **Y** for yes or an **N** for no in the flag field following the specific option. If something else is entered in the flag field, the following error message is displayed:

```
== STATISTICS/TRACE OPTIONS FLAGS SET BY 'Y' OR 'N' ==
```

These Stats and Trace options correspond to the **TRNSTAT** and **TRNTRAC** parameters of the **T09MLSTN** macro, described in **Appendix A – CONFIGURATION STATEMENTS**.

These are the Transaction-Level trace options - to set the API-Level trace options, you must return to the **Listen/Receive Tool** panel and use **PF2=Tool Trace**.

Validation Prompt

Once all fields have been edited successfully, you are prompted to confirm the Add or Alter of this Listen/Receive tool with the following message:

```
PARAMETERS ARE VALIDATED - USE PF9 TO CONFIRM ADD/ALTER
```

You use **PF9** to complete the save of this new tool definition, and you will be returned to the **Listen/Receive Tool** panel for the definition just saved.

If you want to abort the save, and not make any updates, you use **PF3** to return to the previous panel, as shown by the PF key option:

PF3=goback (nosave)

Listen Tool API Trace Options

To dynamically set or remove API-level trace options for the Listen/Receive tool, the **PF2** key is used from the **Listen/Receive Tool** panel as shown by PF key option:

PF2=Tool Trace

This brings up a display very similar to the **Online Trace Options** panel, but the **CPT API Trace Options** panel is used only to generate Transient Data trace records for this tool.

04/27/94 19:02:11	CPT API TRACE OPTIONS TOOL: SEND PORT: 3380	TT01
----------------------	--	------

API STATS==>

CONN: Y TERM: Y

API TRACET==>

ENTR: Y	ARGS: Y	RECV: Y	SEND: Y	TERM: Y	PASS: Y
TOKN: Y	STOR: Y	TPL: Y	TERR: Y	CLOS: Y	CLTD: _

RLSE: _

USE 'Y' TO SELECT, 'N' TO TURN OFF OPTIONS

PF1=? PF2=refresh PF3=goback (nosave) PF9=confirm/goback

Selecting Tool API Trace Options

This panel is used to dynamically set and remove the API trace options for this **Listen/Receive** tool. Setting these options will cause trace records to be written to the Transient Data queue for trace logging, ACTR. These API-level trace options are exactly the same as the options specified on the APISTAT and APITRAC parameters of the T09MLSTN macro that defines this tool.

There is no option to select the Online Trace from this panel. If you want to use the Online Trace on this tool, then you would select either the Listener or Receiver task from the **Browse Detail** panel, just as for any CPT endpoint. Then set the Online Trace options from there, as described in **Online Trace Options Panel** on page 4-21. Use **PF9** to confirm these trace option updates and return to the **Listen/Receive Tool** panel, or use **PF3** to return to that panel without saving the updates.

Send Tool Panel

In the batch definition of the configuration table, the Send Tool parameters are defined by the `T09MSEND` macro. This panel displays those parameters that define the Send Tool as a client task, which connects with a server on a remote host. From this panel, three potential actions can be taken on this tool:

- ◆ Alter the parameters defining the Send Tool.
- ◆ Add a new Send Tool, based on a current tool definition.
- ◆ Set the API-level trace options dynamically for this tool definition.

Examples of these actions will be given after an explanation of the fields of the **Send Tool** panel, shown in this example.

1	04/27/94	2	SEND TOOL	ST	3
	19:04:5	4	CPT INTERFACE STARTED AT 14:04 ON 04/27/94 IN APPLID CICSQA01	P.	1
5	LOCAL HOST NAME: HOMER.INTERLINK.COM. ADDRESS: 138.99.128.13				
	SEND TO WELL-KNOWN PORT: 3383 REMOTE HOST NAME: 138.70.53.17 BUFFERS: 4 @ 1024 BYTES DNR CALLS: YES				
6	SEND TD QUEUE: IPSS TD QUEUE BUFFER: 1024 BYTES TOOL OPTIONS: SEP=CR LF, TRANSLATE USING T09XENG				
	TRAN STATS: TERM TRAN TRACE: ARG, ENTER, TERM, WTD, RTD, XLAT, TOKEN				
	API STATS: CONN, TERM API TRACE: ARG				
7	>				8
9	Pf1=? Pf2=Listen Tools	Pf3=goback Pf4=Utilization	Pf5=Configuration Pf6=Tool Trace	Pf8=Fwd Pf9=Add/alter Send	

ITEM	DESCRIPTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID.
4	The date and time that the CPT Interface was initialized, and the APPLID of this CICS system.
5	The local host name and IP address of the TCP transport provider, to which this CPT Interface is connected.
6	The parameter fields derived from the <code>T09MSEND</code> macro definition.

ITEM	DESCRIPTION
7	The command line used for entering a function code.
8	The message area for command feedback and error notification.
9	The PF keys currently active for this panel.

Send Tool Field Descriptions

Many of the tool screens will appear quite similar at first glance. The key fields to focus on here are the port number and the Remote Host Name, which define where the TCP connection will be made, and the Send TD queue, which defines where the outgoing data will be read.

Local Host Name

The full domain name of the local host's TCP/IP subsystem.

Address

The dotted decimal IP address of the local host's TCP/IP subsystem.



Note: The local host name and address are not specified in the macro definition for the Send tool. These fields are being replicated here from the initial **Configuration Table Parameters** panel for reference purposes only.

Well-Known Port

This port number is where the Send task will connect to the server on the Remote Host.

Remote Host Name

This is the full domain name for the Remote Host, or the dotted decimal IP address of that host.



Note: In the macro definition, the `IPNAME=` parameter can also specify either the dotted IP address of the Remote Host, or its full domain name, which will be translated by DNR into the IP Address.

Buffers

This field defines the number and size of send buffers to be allocated for output data from the connection. 4 buffers at 1024 bytes each is the default.

DNR Calls

This field shows if the Send Tool will automatically issue a DNR call for each new connection it creates to translate the remote server's IP address or domain name.

Send TD Queue

This is the Transient Data queue to which outgoing data is read as TD records.

TD Queue Buffer

This is the maximum record size that can be read from the TD queue. The default is 1024 bytes.

Tool Options

These options define how the TD records will be combined for the output data stream. These options are described in the `T09MSEND` macro definition in **Appendix A – CONFIGURATION STATEMENTS**.

Tran Stats & Trace

These options define what Transaction-Level tracing and statistics are should be written for this tool. The options are described in the `T09MSEND` macro parameters, `TRNSTAT` and `TRNTRAC`, in **Appendix A – CONFIGURATION STATEMENTS**.

API Stats & Trace

These options define what API-Level tracing and statistics are should be written for this tool. The options are described in the `T09MSEND` macro parameters, `APISTAT` and `APITRAC`, in **Appendix A – CONFIGURATION STATEMENTS**.



Note: The Statistics options specify which stats records will be written to the TD queue for statistics, whose default queue name is ACST. The Trace Options specify which trace records will be written to the TD queue for trace logging, whose default queue name is ACTR. The trace records will not be written to the Online Trace, unless a specific trace request is made through that facility.

Warning Messages

This panel will check to see if the Send TD queue specified is defined to CICS. If not, a warning message is displayed.

The parameters displayed for this tool can be modified by hitting the **PF9** key. This will take you to the **Add/Alter Send Tool** panel, described below. That same panel is used to modify the current tool definition or to create a new tool definition, which is why **PF9** has this label at the bottom of the panel:

Pf9=Add/alter Send

Add/Alter Send Tool Panel

This panel provides the ability to modify the parameters of the currently displayed Send tool, or the ability to create a new Send tool definition based on the current display. The key fields that determines if this is an Add or an Alter of the tool are the port number field and the Remote Host field. The initial message displayed on entry to this panel tries to explain this:

CPT SEND TOOL WILL BE ADDED IF NEW PORT OR HOST IS ENTERED

1	04/27/94 19:06:00	2	ADD/ALTER SEND TOOL	3	TA01
4	WELL-KNOWN PORT: 3384 OR SERVICE:				
	RMT HOST IPNAME: 138.70.53.17				
5	SEND BUFFERS: 4 @ 1024 BYTES DNR CALLS:				
	SEND TOOL PARAMETERS				
6	TRANSIENT DATA QUEUE: IPSS BUFFER SIZE: 1024 BYTES				
7	...TRANSLATE TABLE: DEFAULT				
8	...PROCESSING OPTION: SEP=CRLF				
9	STATS OPT==> TRM: Y TRACE OPT==> ENT: N ARG: N RTD: N WTD: N TRM: N TKN: N XLA: N				
	10 PARAMETERS ARE VALIDATED - USE PF9 TO CONFIRM ADD/ALTER				
	Pf1=? Pf3=goback (nosave)				
11	Pf2=refresh Pf9=tool save				

ITEM	DESCRIPTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID.
4	The remote port number and host name that identify this send tool.
5	The Send buffer number and size.
6	The Send Transient Data queue name and its maximum record size.
7	The Translate Table parameter is used to specify a unique ASCII/EBCDIC translation table for this tool.
8	The Processing Options field is used to specify how the TD records are processed to create the output datastream.
9	Stats and Trace options are set on and off dynamically for this Send tool.
10	The message area for command feedback and error notification.

ITEM	DESCRIPTION
11	The PF keys currently active for this panel.

Add/Alter Tool Field Descriptions

Since the Send tool is uniquely defined by the host address and port number it connects to, by changing the Remote Host or port number on this screen, you are implying that you want to Add a new tool definition that will connect to that Host and Port. Of course, you will probably also want to change the Transient Data queue name from which the Send task will read its outgoing records, and perhaps also the record separation parameters defined for that queue.

Port Number

The port number for a new tool can be explicitly entered, or a Service name can be entered, which will be translated to a port number by the Domain Name Resolver. The port number will be edited, and must be a number between 1 and 65,534, since the remote application is a server transaction. Otherwise, the following error message is displayed:

```
== PORT NUMBER MUST BE BETWEEN 1 AND 65534==
```

Remote Host Address

The Remote Host Address can be entered as a full domain name or a dotted decimal IP address, but unfortunately no editing is done on the format of the name or address. So if a mistake is made in your specification, you will create a new Send tool for that host, but you won't discover the problem until the first attempt to connect is rejected.

Send Buffers

The Send Buffer number and size are edited for numerics, and have a total buffer size limit of 61K, or the following error message appears:

```
== BUFFERS * SIZE MUST NOT BE MORE THAN 61K ==
```

DNR Calls

Specify **no** in this field to suppress the automatic DNR call made by the Send Tool for each new connection. **yes** is the default value. Any other value will display the following error message:

```
== DNR CALL OPTION MUST BE 'YES' OR 'NO' ==
```

200-124030-200100

Send Transient Data Queue

The Transient Data queue name is probably the most important field to set besides the Host address and port number. However, if you are adding a new tool, you must make sure that the TD queue name has already been defined in the CICS Destination Control Table, or you will get the following error message:

WARNING: TRANSIENT DATA QUEUE YYYY IS NOT DEFINED TO CICS



Note: There should only be one Send tool reading from the Send TD queue. If multiple record types need to go to the remote service, remember that you can have more than one Send tool client connect to the remote server. So just write each unique record type to its own Send TD queue.

Buffer Size

The Buffer Size of the Transient Data queue defines the maximum record size that can be read from to the specified queue. If processing option, `FILE`, is specified, that means the entire output datastream for the connection must fit into one record. So this buffer must as large as the largest data transmission expected. This field is edited for numerics, and the CICS maximum record size for transient data records is 32,767 bytes.

Translate Table

The Translate Table parameter is used to specify a unique ASCII/EBCDIC translation table for this tool. Otherwise, the default table specified in the `T09MCICS` macro will be used. There is no editing done on this panel field. The two common entries for this field are:

- ◆ `RAW` – No translation is done on the input data stream.
- ◆ `DEFAULT` – The default translation module, `T09XENG`, will be used.

Processing Options

The Processing Options field is used to specify how the TD records will be processed to create the output datastream. One of the four basic options defined by the `OPTIONS` parameter of the `T09MSEND` macro should be specified:

- ◆ `LL` – A length field will be used to separate the outbound records.
- ◆ `SEP` – A separator character will separate the outbound records.
- ◆ `FILE` – A new connection will be open and closed for each TD record.
- ◆ `ALL` – All the TD records will be merged together and sent as one output datastream with no record boundaries indicated.

If option `SEP` is specified, the specific `SEP` characters are defined following an equal sign, just as the `SEP=` parameter is specified in the `T09MSEND` macro. The default specification is `SEP=CRLF`, which means the outgoing records are

separated by ASCII Carriage Return and Line Feed characters, which are hex '0D0A'.

See the T09MSEND macro in **Appendix A – CONFIGURATION STATEMENTS** for other valid SEP= specifications. If an invalid parameter option is specified in this field, you will also be referred to that macro definition in Appendix A by this error message:

```
== ERROR DETECTED IN OPTIONS - SEE MANUAL FOR SPECIFICATIONS
```

Stats and Trace Options

The Stats and Trace Options are set on and off dynamically for this Send tool by entering either a 'Y' for YES or an 'N' for NO in the flag field following the specific option. If something else is entered in the flag field, the following error message is displayed:

```
== STATISTICS/TRACE OPTIONS FLAGS SET BY 'Y' OR 'N' ==
```

These Stats and Trace options correspond to the TRNSTAT and TRNTRAC parameters of the T09MSEND macro, described in **Appendix A – CONFIGURATION STATEMENTS**.

These are the Transaction-Level trace options - to set the API-Level trace options, you must return to the Send tool display and use PF6=Tool Trace.

Validation Prompt

Once all fields have been edited successfully, you are prompted to confirm the Add or Alter of this Send tool with the following message:

```
PARAMETERS ARE VALIDATED - USE PF9 TO CONFIRM ADD/ALTER
```

You use **PF9** to complete the save of this new tool definition, and you will be returned to the **Send Tool** panel for the definition just saved.

If you want to abort the save, and not make any updates, you use **PF3** to return to the previous panel, as shown by the PF key option:

```
PF3=goback (nosave)
```

Send Tool API Trace Options

To dynamically set or remove API-Level Trace Options for the Send tool, the **PF6** key is used from the Send tool display panel as shown by PF key option:

PF6=Tool Trace

This brings up a display very similar to the **Online Trace Options** panel, but the **CPT API Trace Options** panel is used only to generate Transient Data trace records for this tool.

```

04/27/94                                CPT API TRACE OPTIONS                                TT01
19:02:11                                TOOL: SEND PORT: 3380

STATS OPT==>

      CONN: Y      TERM: Y

TRACE OPT==>

      ENTR: Y      ARGS: Y      RECV: Y      SEND: Y      TERM: Y      PASS: Y
      TOKN: Y      STOR: Y      TPL: Y      TERR: Y      CLOS: Y      CLTD: _

      RLSE: _

                                USE 'Y' TO SELECT, 'N' TO TURN OFF OPTIONS

Pf1=?      Pf2=refresh      Pf3=goback (nosave)                                Pf9=confirm/goback

```

Selecting Tool API Trace Options

This panel is used to dynamically set and remove the API trace options for this **Send** tool. Setting these options will cause trace records to be written to the Transient Data queue for trace logging, ACTR. These API-level trace options are exactly the same as the options specified on the APISTAT and APITRAC parameters of the T09MSEND macro that defines this tool.

There is no option to select the Online Trace from this panel. If you want to use the Online Trace on this tool, then you would select the Send task for the **Browse Detail** panel, just as for any CPT endpoint. Then set the Online Trace options from there, as described in **Online Trace Options Panel** on page 4-21.

Use **PF9** to confirm these trace option updates and return to the **Send Tool** panel, or use **PF3** to return to that panel without saving the updates.

Operations

The Operations component of the Administrator Interface provides a collection of functions to diagnose and manage CPT. The CPT Task-Related User Exit (TRU Exit) provides the call interface to the TCP transport provider address space. This Task-Related User Exit must be enabled to CICS before any transaction can execute a CPT call, and when it is disabled to CICS, all TCP and UDP endpoints are terminated.

This Task-Related User Exit is referred to in this section as “the CPT Interface.” The process of enabling the CPT TRU Exit is referred to as “CPT Startup,” and the process of disabling the CPT TRU Exit is referred to as “CPT Shutdown.”

Operations Control Menu Panel

The **Operations Control Menu** panel provides functions to diagnose and manage the CPT Interface. These functions are summarized first for the menu, and then the panels they invoke are described.

This **Operations Control Menu** panel is invoked by the **O** function code from anywhere within the Administrator Interface.

This is an example of the **Operations Control Menu** panel:

04/27/94

OPERATIONS CONTROL MENU

OC01

19:10:05 CPT INTERFACE STARTED AT 14:04 ON 04/27/94 IN APPLID CICSQA01

PF6) WE: WATCH ERROR LOG

PF7) QI: QUIESCE IDLE LISTENERS

PF8) QD: DRAIN CPT INTERFACE

PF9) SD: SHUTDOWN ALL IMMEDIATE

PF10) ZS: RESET ALL STATISTICS

PF11) RT: RESET ONLINE TRACE QUEUE

PF12) RE: RESET ERROR LOG QUEUE

>

Pf1=?

Pf3=goback

Pf4=Utilization

Pf5=Configuration

200
124030-200100

Operations Functions

These operation functions can be selected with the PF key shown before it on each line of the menu, or they can be selected by entering the two-character function code shown after the PF key. These function codes can also be entered on other Administrator Interface panels to jump to these operation functions directly without going through this menu panel.

Watch Error Log

This function lets the administrator browse the CPT error messages that are normally sent to the ACER Transient Data queue. These messages are displayed by a wrap-around queue of Temp Storage records similar to the Online Trace queue.

Quiesce Idle Listeners

This function lets the administrator close the endpoints of all Listener tasks that are currently idle. An idle Listener task is one that currently has no active connections established to its local port number.

Drain CPT Interface

This function lets the administrator close the endpoints of all Listener tasks to prevent a new connection being accepted from a remote host, and it prevents any CICS task from making a connect call to a remote host. Draining the CPT Interface implies that all existing connections are allowed to complete, but establishing a new connection is blocked, in preparation for CPT shutdown.

Shutdown All Immediate

This function lets the administrator force an abortive close on all active connections, and invoke the CPT termination routine, just the same as the IPPR transaction. This action shuts down the CPT interface immediately.

Reset All Statistics

This function lets the administrator reset both the cumulative statistics and the endpoint statistics that are shown on the **Browse Detail** and **Utilization Summary** panels.

Reset Online Trace Queue

This function lets the administrator release the Online Trace if it is being held by a particular terminal, and clears all trace records from the Temp Storage queue. This function can also change the number of records that the Online Trace queue can hold.

Reset Error Log Queue

This function lets the administrator clear all error records from the Temp Storage queue used by the Watch Error Log. This function can also change the number of records that the Error Log queue can hold.

CPT Interface Error Log Panel

The **CPT Interface Error Log** panel displays the CPT error messages normally sent to the ACER Transient Data queue, which is the CPT Error Log. These error messages are copied into a Temp Storage queue where they can be browsed online. This allows an administrator to diagnose CPT error conditions more rapidly by viewing online the messages produced by the CPT Interface.

This is an example of the **CPT Interface Error Log** panel:

```

1 04/27/94                                2 CPT INTERFACE ERROR LOG                                WE01 3
19:17:20

4 00017 CPT125I T09CINIT CPT SUCCESSFULLY COMPLETED DNR REQUEST
**79) CPT111I T09CINIT CPT DEFAULT TRANSLATION TABLE LOADED
00024 CPT236E T09CLSTN TBIND FAILED, DC=0C0E0C16 PROTOCOL ADDR=00010D3400000000
5 **81) CPT226E T09TLSTN CPT LISTEN ROUTINE FAILED, RC=00000021 DC=0C0E0C16
00076 CPT415E T09CLSTN LISTEN PURGED BY TCLOSE
**83) CPT400E T09CLSTN API TPL ERROR; HEX DUMP OF TPL FOLLOWS
**84 7 0000 00092084 EA0A00C8 0473B770 00000000 80000000 080B140A 00000000 0000
**85) 00 8 000920A4 00091F50 00000008 00083A14 FFFFFFFE 00000000 00000000
**86) CPT106I T09CCLOS CPT API ENDPOINT 00091EB8 CLOSED
00141 CPT106I T09CTERM CPT API ENDPOINT 00091EB8 CLOSED
00209 CPT106I T09CTERM CPT API ENDPOINT 000A1138 CLOSED
00036 CPT106I T09CEXIT CPT API ENDPOINT 000A1138 CLOSED
00219 CPT106I T09CTERM CPT API ENDPOINT 000A19B8 CLOSED
00046 CPT106I T09CEXIT CPT API ENDPOINT 000A19B8 CLOSED
00045 CPT413E T09CRECV DISCONNECT INDICATION RECEIVED.
**93) CPT106I T09CEXIT CPT API ENDPOINT 000A1578 CLOSED

9 >
Pfl=?
11 Pf2=Show First Pf6=Window Left/right Pf9=Show Last Pf12=Watch Release
Pf7=Bwd Pf8=Fwd

```

ITEM	ACTION
1	The current date and time on this CICS system.
2	The title of this panel.
3	The panel ID.
4	The CICS task number of the endpoint task generating the traces messages.
5	A line count of subsequent messages produced by the same task.
6	The CPT message number for each new trace message.
7	The hexadecimal displacement field for each line of a hex dump message.
8	The hexadecimal address field for each line of a hex dump message.
9	The command line used for specifying an function code.
10	The message area for command feedback and error notification.
11	The PF keys currently active for this panel.

Error Log Queue Description

The Error Log queue is a Temp Storage queue whose size is determined by the `AISELOG=` parameter of the `T09MCICS` macro specified in the configuration table. The default size of the Error Log queue is 30 records, and its maximum size is 999 records. When the Error Log queue becomes full, the error messages wrap around and begin overlaying the first messages in the queue.

This queue can be changed in size or cleared by the **RE** (Reset Error) function on the **Operations Control Menu** panel. If the Error Log queue is set to zero, this error message displays on the **CPT Interface Error Log** panel:

```
>          ...NO ERROR RECORDS LOGGED...
```



Note: The Error Log queue contains only records from the ACER Transient Data destination. Statistics records sent to ACST and Trace records sent to ACTR destinations are not displayed here, even though these records may be mixed together in a CICS sysout dataset.

Browsing the Error Log Queue

The PF key selections let the user browse forward and backward through the messages, scroll the display left and right, and jump to the first or last message in the queue. When finished with browsing the error messages, you can use the **PF12** key to exit the **CPT Interface Error Log** panel. The **PF12** key releases the Error Log queue, and goes to the **Utilization Summary** panel.

When more than one user is browsing the Error Log queue, only the first user will have the **PF12** option to release the Error Log queue. The other users will have to use the **O** function code to return to the **Operations Control** menu.



Note: New error messages produced while the **CPT Interface Error Log** panel is displayed cannot be added to the Error Log queue. In order to see if any new error messages have arrived, **PF12** must be used to release the Error Log queue and exit the panel. Then the **WE** function can display the **CPT Interface Error Log** panel again.

If the CICS system is brought down and then warm-started, the Temp Storage queue containing the Error Log messages may be kept. The administrator should be aware that error messages displayed on the **CPT Interface Error Log** panel could be from a previous execution of the CICS system, if the Error Log queue has not been reset with the **RE** function.

Quiesce Idle Listeners Function

Entering the **QI** function code, or using **PF7** on the **Operations Control Menu** panel causes the **Quiesce/Close Confirmation** panel to display, prompting for a confirmation of the request with this message:

```
PLEASE USE PF9 (QV) TO CONFIRM REQUEST TO QUIESCE IDLE TASKS!
```

By using the **PF9** key or entering the **QV** function code, the administrator confirms that all idle Listeners tasks will have their endpoints closed and tasks terminated. An idle Listener task is one that currently has no active connections established to its local port number.

If the quiesce request is to be aborted by not confirming on the **Quiesce/Close Confirmation** panel, you can use **PF3** to return to the **Operations Control Menu**, or you can use any of the standard function codes or PF keys to go to another panel.

Drain CPT Interface Function

Entering the **QD** function code, or using **PF8** on the **Operations Control Menu** causes the **Quiesce/Close Confirmation** panel to display, prompting for a confirmation of the request with this message:

PLEASE USE PF9 (QV) TO CONFIRM REQUEST TO DRAIN THE CPT INTERFACE!

By using the **PF9** key or entering the **QV** function code, the administrator confirms that no new connections can be started by either local CICS tasks or from clients on remote hosts. This function causes all Listener tasks to have their endpoints closed and tasks terminated, in preparation for terminating the CPT Interface.

If the drain request is to be aborted by not confirming on the **Quiesce/Close Confirmation** panel, you can use **PF3** to return to the **Operations Control Menu**, or you can use any of the standard function codes or PF keys to go to another panel.

Shutdown All Immediate Function

Entering the **SD** function code, or using **PF9** on the **Operations Control Menu** panel causes a single prompt to display on the **Operations Control Menu**, prompting for a verification of the shutdown with this message:

PLEASE USE PF9 (SV) TO CONFIRM COMPLETE SHUTDOWN REQUEST!

By using the **PF9** key or entering the **SV** function code, the administrator confirms that the CPT Interface is to be terminated and that any remaining connection endpoints will have an abortive close. After entering the confirmation, the next message displayed should report the termination of the CPT Interface:

CPT120I T09TTERM CPT TERMINATION SUCCESSFUL

If the shutdown request is to be aborted by not confirming on the **Operations Control Menu**, you can use **PF3** to return to the **Main Menu** panel, or you can use any of the standard function codes or PF keys to go to another panel.

Reset All Statistics Function

Entering the **ZS** function code, or using **PF10** on the **Operations Control Menu** causes the Reset Statistics panel to display, prompting for a confirmation of the request with this message:

PLEASE USE PF9 (ZV) TO CONFIRM REQUEST TO RESET STATISTICS

By using the **PF9** key or entering the **ZV** function code, the administrator confirms that both the Cumulative Statistics and the Endpoint Statistics being collected for all endpoints will be cleared.

If the reset request is to be aborted by not confirming on the **Reset Statistics** panel, you can use **PF3** to return to the **Operations Control Menu** panel, or you can use any of the standard function codes or PF keys to go to another panel.

Reset Online Trace Queue Function

Entering the **RT** function code, or using **PF11** on the **Operations Control Menu** causes the **Reset Trace/Error Online Queue** panel to display, prompting for a new queue size and a confirmation of the request with these messages:

ENTER NUMBER 0,30-999 TO CHANGE SIZE OF T/S QUEUE FOR TRACE

AND/OR

USE PF9 TO CONFIRM REINITIALIZATION OF THE QUEUE

Normally, just a confirmation is given with **PF9**. The next screen displayed will be the **Utilization Summary** panel, with this message:

> TRACE ONLINE QUEUE HAS BEEN RESET

Changing Online Queue Size

If a new queue size is typed on the command line and **Enter** is used instead of **PF9**, then no reset is done and the **Utilization Summary** panel displays with this message:

> TRACE QUEUE SIZE WILL CHANGE AT NEXT INITIALIZATION

If a new queue size is typed on the command line and **PF9** is used to confirm the reset, then the **Utilization Summary** panel displays with this message:

> TRACE ONLINE QUEUE HAS BEEN RESET AND SIZE CHANGED

The Online Trace queue cannot be set to less than 30 entries or more than 999 entries. If it is set to zero entries, the Online Trace is turned off and this message displays on the **Trace Options** panel used to acquire the Online Trace:

ONLINE TRACE NOT AVAILABLE

If the reset request is to be aborted by not confirming on the **Reset Trace/Error Online Queue** panel, you can use **PF3** to return to the **Operations Control Menu** panel, or you can use any of the standard function codes or PF keys to go to another panel.

Reset Error Log Queue Function

Entering the **RE** function code, or using **PF12** on the **Operations Control Menu** causes the **Reset Trace/Error Online Queue** panel to display, prompting for a new queue size and a confirmation of the request with these messages:

ENTER NUMBER 0,30-999 TO CHANGE SIZE OF T/S QUEUE FOR ERROR

AND/OR

USE PF9 TO CONFIRM REINITIALIZATION OF THE QUEUE

Normally, just a confirmation would be given with **PF9**. The next screen displayed is the **Utilization Summary** panel with this message:

```
> ERROR ONLINE QUEUE HAS BEEN RESET
```

Changing Error Log Queue Size

If a new queue size is typed on the command line and **Return** is used instead of **PF9**, no reset will be done and the **Utilization Summary** displays with this message:

```
> ERROR QUEUE SIZE WILL CHANGE AT NEXT INITIALIZATION
```

If a new queue size is typed on the command line and **PF9** is used to confirm the reset, the **Utilization Summary** displays with this message:

```
> ERROR ONLINE QUEUE HAS BEEN RESET AND SIZE CHANGED
```

The Error Log queue cannot be set to less than 30 entries or more than 999 entries. If it is set to zero entries, the Error Log is turned off and this message displays on the **CPT Interface Error Log** panel used to display the Error Log:

```
...TRACE NOT REQUESTED -OR- QUEUE WAS RESET
```

If the reset request is to be aborted by not confirming on the **Reset Trace/Error Online Queue** panel, you can use **PF3** to return to the **Operations Control Menu Panel**, or you can use any of the standard function codes or PF keys to go to another panel.

Startup CPT Interface Function

The Startup function for CPT is not normally displayed on the **Operations Control Menu**, because the CPT Interface is usually active when using it. However, if the CPT Interface was terminated or did not initialize successfully, this message displays on the **Main Menu** for the Administrator Interface:

```
CPT INTERFACE NOT AVAILABLE (CICS 'AEY9')
```

If the **O** function code or **PF6** is used on the **Main Menu** at that time, a special version of the **Operations Control Menu** is displayed to start up the CPT Interface, as shown in this example:

04/27/94 19:22:08	OPERATIONS CONTROL MENU	OC01
PLEASE USE PF9 (SU) TO STARTUP CPT		
> Pf1=? Pf3=goback Pf4=Utilization Pf5=Configuration		

This version of the **Operations Control Menu** has only a single function to confirm the startup of the CPT Interface. By using the **PF9** key or entering the **SU** function code, the administrator confirms that the CPT Interface is to be initialized. This action is equivalent to using the IPST transaction to start the CPT Interface.

The next panel displayed will be the **Utilization Summary** panel. The administrator can just use **Enter** on the **Utilization Summary** panel to see if the Listener tasks defined in the configuration table have started up. Then you can use the **WE** function to go to the **CPT Interface Error Log** panel to check the CPT initialization messages that are written to the CPT Error Log.



Chapter 5

CPT Operations

This chapter describes how to manage the operation of the CPT Interface. It includes these sections:

- ◆ Overview
Introduces the CPT Operations task.
 - ◆ Initialization
Describes the methods available for initializing the CPT Interface and for diagnosing initialization failure.
 - ◆ Termination
Describes the methods available for terminating the CPT Interface and for diagnosing termination failure.
 - ◆ Transaction Security
Describes how to secure CPT initialization and termination.
 - ◆ Cumulative Statistics Records
Describes how to collect cumulative statistics of CPT transactions.
-

200801-024030-200100

200-24030-200100

Overview

This section describes the details of the initialization and termination of the CPT Interface, and how these actions can be executed automatically by using the CICS Program List Tables (PLT). The CPT Interface can also be started and stopped manually any number of times without having to bring down the CICS system.

This capability to re-initialize the CPT Interface allows a new Configuration Table module to be assembled and used with new parameters that could not be modified through the Administrator Interface. This section also describes how to secure the startup and shutdown functions, so that they can only be executed by authorized users.

CPT Initialization

The CPT Interface is initialized by program, `T09TSTRT`, which has a default transaction ID of IPST. This program can be invoked in three different ways to initialize the CPT Interface.

- ◆ Specify the `T09TSTRT` program as an entry in the CICS PLT table for program initialization (PLTPI). This will automatically startup the CPT Interface as part of CICS initialization.
- ◆ Enter the IPST transaction from a CICS terminal. This manual startup of the CPT Interface is useful on test systems, where the CPT Interface can be brought up and down manually to test a new Configuration Table. If a new copy of the `T09CONFIG` module was assembled, then the `T09TSTRT` program will automatically load the newest version of that load module from its DFHRPL load library.
- ◆ Invoke CPT startup from the Administrator Interface. Enter the IPAO transaction to display the **Operations Control** menu, and then use **PF9** to confirm the startup. This method will start the IPST transaction, and then immediately display the **Utilization Summary**.

Regardless of which startup method is used, the `T09TSTRT` program will perform the same set of tasks to initialize the CPT environment:

- ◆ Enable the CPT Task-Related User Exit program, `T09COMON`.
- ◆ Load the CPT Configuration Table module, `T09CONFIG`.
- ◆ Open the Transient Data queues for Error, Statistics, and Trace, whose default names are ACER, ACST, and ACTR.
- ◆ Connect to the address space of the TCP transport provider, whose subsystem ID is specified by the `SSN=` parm in the Configuration Table.
- ◆ Complete a DNR request with the DNR service provider, whose subsystem ID is specified by the `DNRSSN=` parm in the Configuration Table.
- ◆ Load the CPT Translation Table module, whose default name is `T09XENG`.
- ◆ Start all the Listener tools specified in the Configuration Table.

CPT Initialization Messages

As these tasks are accomplished, CPT startup messages will be written out to the system console log. The following set of messages indicate a successful CPT initialization:

```
+CPT100I T09TSTRT CPT TRUE EXIT INTERFACE ENABLED
+CPT101I T09CINIT CPT ESTABLISHED API SESSION FOR SUBSYSTEM ACSS
+CPT125I T09CINIT CPT SUCCESSFULLY COMPLETED DNR REQUEST TO ACSS
+CPT111I T09CINIT CPT DEFAULT TRANSLATION TABLE LOADED
+CPT109I T09TSTRT CPT INITIALIZATION SUCCESSFUL
```

The 'ACSS' subsystem ID shown in these messages is the default name for the SNS/TCP transport provider, and may be different for your installation.

There are also a set of messages written out to the CPT Transient Data queues at CPT initialization. If all three queues are routed to the same CICS sysout dataset, then these messages will appear together in the CICS sysout to indicate a successful CPT initialization:

```
00022 CPT123I T09CINIT INITIAL WRITE TO ERROR LOG TD QUEUE
00022 CPT810I T09CINIT INITIAL WRITE TO STATISTICS LOG TD QUEUE
00022 CPT936I T09CINIT INITIAL WRITE TO TRACE LOG TD QUEUE
00022 CPT101I T09CINIT CPT ESTABLISHED API SUBSYSTEM SESSION
00022 CPT125I T09CINIT CPT SUCCESSFULLY COMPLETED DNR REQUEST
00022 CPT111I T09CINIT CPT DEFAULT TRANSLATION TABLE LOADED
```

The '00022' preceding these messages is the CICS task number of the IPST startup transaction performing the initialization.

CPT Initialization Failure

If a CPT/TOOL transaction or a user-written application using the CPT/API services gets an AEY9 abend, it is an indication that the CPT Interface has not been started, or that the initialization has failed. AEY9 is a standard CICS abend issued when an application call is made to a Task-Related User Exit that has not been enabled.

There are several reasons why CPT initialization might fail. First, it is necessary that the TCP address space be up and running with its API active when the CPT initialization is started. It must be the TCP transport provider specified by the SSN= and DNRSSN= parms of the current Configuration Table. If the TCP API is not available, CPT initialization will fail with the following messages sent to the console log:

```
+CPT100I T09TSTRT CPT TRUE EXIT INTERFACE ENABLED
+CPT102I T09CINIT CPT INITIALIZATION PENDING API SUBSYSTEM OPEN
+CPT105E T09CINIT CPT INITIALIZATION FAILED FOR
      SUBSYSTEM-xxxxyyyy
+CPT114E T09CINIT CPT INITIALIZATION ROUTINE FAILED RC= 18
+CPT132E T09CINIT CPT TRUE EXIT INTERFACE DISABLED
+CPT105E T09TSTRT CPT INITIALIZATION FAILED
```

The xxxx and yyyy subsystem IDs shown in message CPT105E are the TCP and DNR provider subsystems, as specified by parms SSN=xxxx, and DNRSSN=yyyy.

Another reason for CPT initialization failure is when the Task-Related User Exit cannot be enabled. This would occur if CICS could not load the TRU Exit module, T09COMON, or the Configuration Table module, T09CONFIG, or the Translation Table module, T09XENG, or user-provided alternate translation table. This might be caused if the actual load module was missing from a DFHRPL loadlib, or if a problem existed with its PPT entry, such as having the module currently marked DISABLED.

Finally, CPT initialization can fail if any of the three Transient Data queues specified by the QNAMES= parm in the Configuration Table are not defined in the CICS DCT or cannot be opened. In all of these cases, error messages will be sent to the console syslog, concluding with the CPT INITIALIZATION FAILED message.

CPT Termination

The CPT Interface is terminated by program, `T09TTERM`, which has a default transaction id of IPPR. This program can be invoked in three different ways to shutdown the CPT Interface.

- ◆ Specify the `T09TTERM` program as an entry in the CICS PLT table for system shutdown (PLTSD). This will automatically shutdown the CPT Interface as part of CICS termination.
- ◆ Enter the IPPR transaction from a CICS terminal. This manual shutdown of the CPT Interface is useful on test systems, where the CPT Interface can be brought down manually, perhaps to load a new copy of the Configuration Table module, `T09CONFIG`.
- ◆ Invoke CPT termination from the Administrator Interface. Enter the IPAO transaction to display the **Operations Control** menu. Then enter the **SD** function or use **PF9** to select Shutdown All Immediate from the menu, and use **PF9** again to confirm the shutdown.

The Administrator Interface now provides a more graceful method of terminating the CPT Interface. Before selecting the Shutdown All Immediate function, the CPT interface can be drained, which allows all existing connections to complete first. On the **Operations Control** menu, selecting the **QD** function to Drain CPT Interface will block the initiation of any new connections from a remote host or from a local CICS transaction. But CPT will still allow all existing connections to complete and do an orderly close.

In previous releases of CPT, the `T09TTERM` termination program would always wait for active connections to close. But now that the drain function is available in the Administrator Interface, the `T09TTERM` program can be more ruthless, by issuing an abortive close on any TCP or UDP endpoints that are still active when it is invoked.



Note: This immediate shutdown assures that CPT termination via the shutdown PLT will not impede the CICS system shutdown.

Regardless of how it is invoked, the `T09TTERM` program will perform the same set of tasks to terminate the CPT environment:

- ◆ Issue an abortive close on all TCP and UDP endpoints still active.
- ◆ Disconnect from the address space of the TCP transport provider, whose subsystem ID is specified by the `SSN=` parm in the Configuration Table.
- ◆ Release the CPT Translation Table module, `T09XENG`.
- ◆ Disable the CPT Task-Related User Exit program, `T09COMON`.
- ◆ Release the CPT Configuration Table module, `T09CONFIG`.

CPT Termination Messages

As these tasks are accomplished, CPT termination messages will be written out to the system console log. The following set of messages indicate a successful CPT termination:

```
+CPT103I T09CTERM CPT TERMINATED API SUBSYSTEM SESSION
+CPT132E T09TTERM CPT TRUE EXIT INTERFACE DISABLED
+CPT120I T09TTERM CPT TERMINATION SUCCESSFUL
```

There are also a set of messages written out to the CPT Transient Data queues at CPT termination. The amount of messages prior to CPT901I may vary depending on the number of endpoints that are forced to close at CPT termination.

```
00066 CPT413E T09CCONN DISCONNECT INDICATION RECEIVED.
00066 CPT234E T09CCONN TCONFIRM FAILED,DC=08030C40 PROTOCOL
      ADDR=...
00030 CPT415E T09CLSTN LISTEN PURGED BY TCLOSE.
00085 CPT901I MODULE T09TTERM 04/12/94 09.54 A1011449 ENTERED AT
      ...
00071 CPT106I T09CEXIT CPT API ENDPOINT 000A2008 CLOSED
00085 CPT106I T09CTERM CPT API ENDPOINT 00095888 CLOSED
00085 CPT103I T09CTERM CPT TERMINATED API SUBSYSTEM SESSION
00085 CPT112I T09CTERM CPT DEFAULT TRANSLATION TABLE RELEASED
00085 CPT130E T09TTERM CPT PROGRAMMER INTERFACE DISABLED
00085 CPT131E T09TTERM CPT CONFIGURATION TABLE T09CONFG RELEASED
```

The '000nn' preceding these messages is the CICS task number of the CICS transaction that had these CPT messages issued on its behalf.

TCP Transport Provider Failure

If the address space for the TCP transport provider fails or is shut down, then the CPT termination program, T09TTERM, will be automatically invoked. All application endpoints will be abortively closed, just as if the IPPR transaction had been executed. The messages sent to the console log and TD queues will be the same as when the termination program is invoked by transaction or by the Administrator Interface.

It is the responsibility of the application program to handle the return codes for the immediate disconnects that could be returned from its last CPT call, indicating that the CPT Interface is terminating. The application should rollback any committed resources if it had a syncpoint established. The CPT/TOOLS will do this automatically when their Transient Data queues are protected. The application should also be ready to handle the AEY9 abend which will occur if the application tries to issue a CPT call when the CPT Interface is down.

CPT Transaction Security

Access to the CPT startup and shutdown functions can be restricted by using normal CICS transaction security. Regardless of whether an external security manager or native CICS security is used, the following CICS transactions should be protected:

- ◆ IPST — CPT Interface Initialization
- ◆ IPPR — CPT Interface Termination
- ◆ IPAO — Administrator Interface Operations Control

If native CICS security is used, this is done by specifying a `TRANSEC=` value greater than 1 in the PCT entry or RDO definition for these transactions. This will require that a terminal user be signed on with that transaction security key before he can execute the CPT startup or shutdown functions.

CPT Cumulative Statistics Records

The Administrator Interface can collect cumulative statistics on the Client or Server ports being used by CICS transactions. These statistics have counts for the total number of connections made to that local or remote port, the total number of messages sent and received on that port, and the cumulative byte length of all those messages. These counts can show the volume of activity for a particular CPT application.

These statistical counts are normally displayed on the **Browse Detail** panels of the Administrator Interface, but they can also be collected for offline batch analysis. To do this, the `AITDSTAT=` parm must be specified on the `T09MCICS` macro in the Configuration Table. This parm specifies the name of a Transient Data queue, which will receive the cumulative statistics records when they are 'discarded' from the Administrator Interface.

The cumulative statistics can be reset at any time in the Administrator Interface by using the **ZS** function on the **Operator Control Menu** panel. If the Statistics Discard queue has been defined, then the currently accumulated statistics will be written out as a separate TD records to this queue before re-initializing the counts. And when the CPT Interface is terminated, these records will be written out again to the queue, so that a history of total activity can be maintained.

These cumulative statistics records are not in a printable format, so this queue should not be routed to a sysout dataset, like the three standard CPT logs. Instead, these records should be sent to a sequential dataset, where they can be collected and later analyzed by a batch program. If you do not intend to do any historical or cumulative analysis of these records, then it is best not to define the Statistics Discard queue at all. Just let the `AITDSTAT=` parm default to a null queue name, which means that the counts will truly be discarded when the statistics are reset or CPT is shutdown.

If you do intend to write a batch analysis program, there is an Assembly language description of the fields in member `T09AITAB` in the CPT macro library.

200801-024030-200100



Appendix **A**

CONFIGURATION STATEMENTS

This appendix describes how to configure the CPT environment and the automated tools. It includes these sections:

- ◆ Overview
Introduces the configuration statements described in this appendix.
 - ◆ CPT Definition Statements
Describes each of the CPT definition statements and how to use them.
-

2008n1-024030-200100

201
J24030-200100

Overview

This appendix describes how to configure the CPT environment and the automated tools. The environmental macro instructions are required, while the automated tool macro instructions are optional.

- ◆ The `T09MCICS` macro instruction defines the CPT environment. This macro defines the communication subsystem, support transaction IDs, default translation table, and log transient data queues to the CICS interface routines. This configuration macro instruction is required.
- ◆ The `LISTEN` tool is selected by defining the `T09MLSTN` macro statement. Each TCP server port can be defined, along with its associated data processing transaction ID. This transaction ID can be the `RECEIVE` tool or a user-written program. The `LISTEN` tool can be defined to let the client determine either the transaction ID to start or the transient data queue in which to write client-specified data.
- ◆ The `RECEIVE` tool is selected by defining the `T09MRECV` macro statement. The `RECEIVE` tool provides a simplex data transfer mechanism for receiving network data. Data received from the network is parsed according to configured options and then written to a transient data queue. The transient data queue name can be configured or dynamically resolved.
- ◆ The `SEND` tool is selected by defining the `T09MSEND` macro statement. The `SEND` tool or client facility provides a simplex data transfer mechanism for transmitting network data. Data is read from a transient data queue and parsed according to configured options before it is transmitted over the network. Additionally, static or dynamic server resolution can be configured.
- ◆ The `T09END` macro statement defines termination of the configuration. It does not need parameter settings. This is an environmental configuration macro instruction and is required.

These macro statements are assembled and link edited into a load module called `T09CONFIG` (whose name cannot be changed) and placed in the CPT load library. The `T09CONFIG` member in the `T09SAMP` library contains the distributed configuration table. While the `T09ASMLK` member in the `CNTL` library contains JCL to assemble and link edit the configuration table.

CPT Definition Statements

This section describes the CPT macros and their parameters, used to define the CPT environment and the CPT tools.

T09MCICS

The **T09MCICS** macro defines the associated TCP communication subsystem, the CICS transaction IDs, and queue names used by the CPT interface. There must be only one of these macros in the configuration, and it must be the first statement.

```
T09MCICS  [SSN=ssn]
          [,DNRSSN=dssn]
          [,TRANSID=(stop[,lstn])]
          [,TRANSTBL=tblname]
          [,QNAME=(stat,trace,error)]
          [,CICSENT=Y:N]
          [,AITABSZ=1000]
          [,AITSTRAC=nnn]
          [,AITSELOG=30]
          [,AITDSTAT=]
          [,AIPREFIX=ccc]
          [,AISTATOP=YES:NO]
```

PARAMETER	DEFINITION
SSN	Describes the subsystem identifier for the CPT routines to use in the TCP subsystem identifier. ◆ <i>ssn</i> – Defines the subsystem identifier of the SNS/TCPaccess address space. This is the <i>SSN</i> parameter on the execution JCL of SNS/TCPaccess. The default is <i>ACSS</i>.
DNRSSN	Describes the subsystem identifier for the CPT routines to use in Domain Name Resolver (DNR) calls. It is generally the same as the <i>SSN=</i> parameter. ◆ <i>dssn</i> – Specifies the subsystem identifier of the SNS/TCPaccess DNR. This defaults to the <i>ACSS</i> for SNS/TCPaccess.
TRANSID	Defines transaction IDs used in CPT operation. ◆ <i>stop</i> – Specifies the transaction ID for the program T09TTERM as defined to CICS. This transaction ID disables the active CPT environment. The default is <i>IPPR</i>. ◆ <i>lstn</i> – Specifies the transaction ID for the program T09TLSTN as defined to CICS. It must be specified if any T09MLSTN macros follow in the configuration. The default is <i>IPTL</i>.
TRANSTBL	Defines a default translation table name for all following configuration statements. This load module must be defined to CICS through the PPT or RDO, or CPT initialization will fail. ◆ <i>tblname</i> – Specifies the name of the default ASCII to EBCDIC and EBCDIC to ASCII translation table for the rest of the configuration. Default: T09XENG

PARAMETER	DEFINITION
QNames	Defines transient data queue names for various classes of messages produced by the CPT routines. Entries must be defined in the CICS DCT for each of the queue names specified, or CPT initialization will fail. ◆ <i>stat</i> – Defines the transient data queue name where statistics information is logged. The default is ACST. ◆ <i>trace</i> – Defines the transient data queue name where trace information is logged. The default is ACTR. ◆ <i>error</i> – Defines the transient data queue name where error messages information is logged. The default is ACER.
CICSENT	Determines whether or not a CICS ENTER TRACE command will be executed every time a CPT service routine is entered. ◆ Y = On ◆ N = Off Default: V
AITABSZ	Administrator Interface Table Size Defines the number of table entries used to hold CPT cumulative statistics for online display. Default: 1000
AITSTRAC	Administrator Interface Temporary Storage for Trace Defines the number of temporary storage records for which to hold online trace entries. The maximum is 999. Zero disables the Online Trace.
AITSELOG	Administrator Interface Temporary Storage for Error Log Defines the number of temporary storage records for which to hold Error Log entries for online display. The maximum is 999. Zero disables the Error Log display. Default: 30
AITDSTAT	Administrator Interface Statistics Destination Defines an optional Transient Data destination to which statistics will be written when CPT is shut down, or when statistics capture is reset. Default: Null - No queue name defined.
AIPREFIX	Administrator Interface Transid Prefix Defines the 3-character prefix of all of the Administrator Interface transactions to avoid conflict with existing Transids. Also used to define the Temporary Storage queue names used by the Administrator Interface. Default: IPA
AISTATOP	Administrator Interface Statistics Option Defines if CPT cumulative statistics are to be collected for online display by the Administrator Interface. Default: No

200801-024030-200100

T09MLSTN

The **T09MLSTN** macros define the ports where the CICS interface listen. When a connection is made, a defined data processing transaction is initiated. Optional buffering specifications, statistic, and tracing option can be defined.

```
T09MLSTN  ,TRANSID=transid
          [,PORT=(number)]
          [,SERVICE=(name)]
          [,PARM=parm,
          [,RCVBUF=(number,size)]
          [,SNDBUF=(number,size)]
          [,TRNSTAT=( [TERM] )]
          [,TRNTRAC=( [ARG] [,ENTER] [,TERM] [,WTD] [,RTD]
                    [,XLAT] [,TOKEN] )]
          [,APISTAT=( [CONN] [,TERM] )]
          [,APITRAC=( [ENTER] [,ARG] [,RECV] [,SEND] [,TERM]
                    [,PASS] [,CLOSE] [,TPLERR] [,TOKEN] [,TPL]
                    [,RLSE] [,STOR] )]
          [,CLNTIME=]
          [,SYNC]
          [,DNR=YES]
```

PARAMETER	DEFINITION
TRANSID	Required keyword if the transaction to process the TCP connection is the only one associated with this LISTEN tool. If the connecting client will determine either the transaction to start or the transient data queue to which to write, then CLNTIME is required. ◆ <i>transid</i> – Defines the transaction to start when a TCP/IP connection is established. Note: If the TRANSID server option is chosen, then PORT or SERVICE must also be specified. Specifying the PARM keyword is optional.
PORT	Defines the transport provider port number to listen on for connections. ◆ <i>number</i> – Defines the transport provider port number to listen on. Note: This field or the service name field is required. PORT and SERVICE are mutually exclusive.
SERVICE	Defines the transport provider service name that represents the port number to listen on for connections. ◆ <i>name</i> – Defines the transport provider service name character string. Note: This field or the port field is required if the TRANSID server option is used. PORT and SERVICE are mutually exclusive.
PARM	Defines the label on an automated RECEIVE tool macro (T09MRECV) statement for this server. ◆ <i>parm</i> – Defines the label on the desired receive tool macro Note: This field is required when using the RECEIVE tool via the TRANSID server option. User-written data processing applications do not require this keyword and value.

PARAMETER	DEFINITION
RCVBUF	Defines buffering space for the CPT API routines. This buffer space is used to retrieve data from TCP/IP. The buffer number and size are multiplied to determine total receive storage allocation. This field is not used by the SEND tool. ◆ <i>number</i> – Specifies the number of input buffers to be allocated when data is to be received. Note: This number should be at least 2 for performance reasons. The default is 4. ◆ <i>size</i> – Specifies the maximum size of an input or receive buffer. Note: The combination of these must not exceed 61K. The default is 1024.
SNDBUF	Defines buffering space for the CPT routines. This buffer space is used to transmit data to TCP/IP. The buffer number and size are multiplied to determine total send storage allocation. Note: The combination of these must not exceed 61K. ◆ <i>number</i> – Specifies the number of output buffers to be allocated when data is to be transmitted. Note: This number should be at least 2 for performance reasons. The default is 4. ◆ <i>size</i> – Specifies the maximum size of an output or send buffer. The default is 1024.
TRNSTAT	Defines statistic logging options for transaction level code of the CPT routines. ◆ TERM – Specifies that a statistic message be generated upon termination of a connection. This message contains transaction data transfer information.
TRNTRAC	Defines tracing options that may be useful in debugging. This defines tracing options for the transaction level (automated tool) code of the CPT routines. ◆ ARG – Specifies that a hex dump of all CPT arguments be generated before and after every CPT service request. ◆ ENTER – Specifies that a message be generated upon entry into a transaction or automated tool program. ◆ TERM – Specifies that a message be generated during task termination processing. ◆ WTD – Specifies that a hex dump of all data be generated before every WRITEQ TD command. ◆ RTD – Specifies that a hex dump of all data be generated after every READQ TD command. ◆ XLAT – Specifies that a hex dump of all data be generated before and after every XLAT service request. ◆ TOKEN – Specifies that a hex dump of the connection token be generated upon establishment of a connection.

200801-024030-200100

PARAMETER	DEFINITION
APISTAT	Defines statistic logging options for the API level code of the CPT routines. ◆ CONN – Specifies that a message be generated upon establishment of either a server or client connection. This message contains protocol address information. ◆ TERM – Specifies that a statistic message be generated upon termination of a connection. This message contains <i>Task-Related User Exit (TRUE)</i> data transfer information.
APITRAC	Defines tracing options that may be useful in debugging. This defines tracing options for the API level code of the CPT routines. ◆ ENTER – Specifies that a message be generated upon entry into a CPT TRUE. ◆ ARG – Specifies that a hex dump of the caller's arguments be generated upon entry and exit of a CPT TRUE. ◆ RECV – Specifies that a hex dump of the transport provider's input (RECEIVE) data be generated. ◆ SEND – Specifies that a hex dump of the transport provider's output (SEND) data be generated. ◆ TERM – Specifies that a message be generated upon exit from a CPT TRUE. ◆ PASS – Specifies that a hex dump of all resources related to a passed (GIVE/TAKE) connection be generated. ◆ CLOSE – Specifies that a hex dump of all resources related to a closed connection be generated. ◆ TPLERR – Specifies that a hex dump be generated of all transport provider parameter list requests that are completed in error. ◆ TOKEN – Specifies that a hex dump of the caller's token be generated upon entry into a CPT TRUE. ◆ TPL – Specifies that a hex dump of all transport provider parameter list requests that are issued be generated. ◆ RLSE – Specifies that a message be generated upon reception of a transport provider release indication. ◆ STOR – Specifies that a hex dump of the argument used for storage management be generated upon entry and exit from the CPT TRUE storage management routine. ◆ CLTD – Specifies that a hex dump of data written to transient data queues be generated when the Client-Data Listener option is used (via the CLNTIME parameter).

20 J24030-200100

PARAMETER	DEFINITION
CLNTIME	Client-Data Listener option Specifies that the Listen call will receive the input datastream to determine the transaction ID to be started. See Client-Data Listener Option for the required input formats. The CLNTIME keyword is mutually exclusive of the TRANSID and PARM keywords. The CLNTIME value indicates the number of seconds the LISTEN tool will wait to receive the input parameter datastream from the client, once the client has successfully established a connection with the LISTEN tool. If CLNTIME is specified, the new trace flag, ACMTCLTD, can be specified to trace the CLNTPARM written to TDQN transient data queue via CPT918I. Default: 0 (No Client-Data Listener option)
SYNC	Listen Syncpoint option Issue SYNCPOINT from LISTEN before spawning a receive transaction. Default: Null (No Syncpoint)
DNR	DNR Suppression option Indicates whether to resolve remote IP addresses into IP names with DNR calls. Specify 'Yes' or 'No'. Default: Yes

Client-Data
Listener Option

The client data specifying the server options can be in any of these formats:

TRAN
TRAN, UUUUUUUUUUUU
TRAN, UUUUUUUUUUUU, IC, HHMMSS
TDQN, UUUUUUUUUUUU, TD
TRAN, , IC, HHMMSS
TDQN, , TD

PARAMETER	DEFINITION
TRAN	A 1- to 4-character transaction ID to start, passing CLNTPARM to the started transaction
UUUUUUUUUUUU	A 1- to 35-byte user data and is passed to the started transaction or written to the transient data queue in the field CLNTDATA.
IC	Specifies that TRAN is to be started in HHMMSS; if left blank, startup is immediate.
HHMMSS	Hours, minutes, and seconds for IC option.
TD	Indicates that CLNTPARM will be written into the transient data queue, TDQN.

CLNTPARM DS 0CL72
TOKEN DS F NEW TOKEN
DS CL16 RESERVED
CLNTDATA DS CL36 UP TO 35 BYTES OF CLIENT
DATA

```
PROTADDR DS 0F
DOMAIN   DS H    FAMILY
RPORT    DS H    REMOTE PORT
RADDR    DS F    REMOTE IPADDR
DC XL8'00'RESERVED
```



Note: Using `CLNTIME` places the listener at risk of being tied up until the client actually sends the server options or until the `CLNTIME` value expires.

T09MRECV

The T09MRECV macros define the inbound simplex data processing options. When the LISTEN tool establishes a connection the RECEIVE tool is initiated. The data from the connection is placed in the CICS transient data queue as specified by either the QNAME keyword or is resolved dynamically from the first four bytes of the data received. The OPTIONS and SEP keywords define how the data is processed before being written to the specified CICS transient data queue. Typically, these queues have a TRIGLEV= parameter coded for them in the DCT, as well as a TRANSID specified for them in a user-written CICS transaction to process the data. The program name for the RECEIVE transaction is T09TREC.V.

```
T09MRECV ,QNAME=name
          [,OPTIONS=( [QNAME]
                      [,TRANS|RAW]
                      [,LL|SEP|ALL|FILE] ) ]
          [,SEP=(CRLF|CR|LF|X'xx..')]
          [,TRANSTBL=tblname]
          [,TDQBUF=(tbuf)]
          [,TRNSTAT=(TERM)]
          [,TRNTRAC=( [ARG] [,ENTER] [,TERM] [,WTD] [,RDT] [,XLAT]
                      [,TOKEN] ) ]
```

PARAMETER	DEFINITION
LABEL	LABEL must match the PARM=VALUE on the associated LISTEN tool.
QNAME	Keyword defining the transient data queue name for the CPT receive transaction to write its data. ◆ <i>name</i> – Defines the CICS transient data QNAME where the data is sent. Note: This keyword cannot be specified if OPTIONS=QNAME is defined.

PARAMETER	DEFINITION
OPTIONS	Defines how data is processed when it is received on a TCP/IP connection. ◆ QNAME – Specifies that the first record be used to determine the CICS Transient Data QNAME. The first four bytes of the record specify the QNAME and are not written to the queue. Note: This option is mutually exclusive with the QNAME=name keyword. ◆ TRANS or RAW – Specifies whether (TRANS) or not (RAW) to translate the data from ASCII to EBCDIC upon receipt from the remote system. Default is. ◆ LL – Specifies that the first two bytes of data is a binary length field that does not include the length field. It has a minimum value of 1 (i.e., there must be one byte of data. The specified amount of data is read from the TCP connection and written to the CICS transient data queue as one record. It has a maximum length of 32769, for 32767 bytes of data. The length field is not written. ◆ SEP – Specifies that the records are separated by characters in the data. The characters that separate the records are defined in the SEP=keyword. The separator sequence is searched for before the data is translated, if translation is requested. ◆ ALL – Specifies that the entire contents of the CICS transient data queue be received as a single TCP session with no indication of record boundaries. ◆ FILE – Specifies that the entire TCP session be received and written to the CICS transient data queue as one record. This limits the amount of data in one TCP session to 32767 bytes. ◆ STRIP – Specifies that the separator sequence not be written to the transient data queue. ◆ NOSTRIP – Specifies that the separator sequence is written to the transient data queue. Default: TRANS and SEP
SEP	Keyword that is only applicable if OPTIONS=SEP. SEP= specifies the characters that separate records. ◆ CRLF – Specifies that records are separated with the ASCII carriage return-line feed character. The ASCII character for CRLF is x'0D0A'. ◆ CR – Specifies that records are separated with the ASCII carriage return character. The ASCII character for CR is x'0D'. ◆ LF – Specifies that records are separated with the ASCII line feed character. The ASCII character for LF is x'0A'. ◆ X'xx...' – Specifies that records are separated with a specified hexadecimal string. The string must be 2 or 4 hexadecimal digits long. Data is received from the TCP connection until a separator sequence is found, then data is written to the CICS transient data queue as one record. The maximum length of one record is 32767. The separator sequence is not translated. Note: If the separator sequence appears anywhere within the data, it is interpreted as a record separator by T09TREC. This may lead to unpredictable results.

20124030-200100

PARAMETER	DEFINITION
TRANSTBL	Keyword defining an alternate translation table to use when processing data on connections to this port. Otherwise, the default translation table specified on the T09MCICS statement is used. ◆ <i>tblname</i> – Specifies the name of the ASCII and EBCDIC translation table for this entry Note: This parameter overrides the default specified in the T09MCICS statement.
TDQBUF	Keyword defining buffering space for the CPT routines to build a transient data queue record. ◆ <i>tbuf</i> – Specifies the size of the allocated transient data queue buffer. This is the maximum amount of data to be contained in one logical record. A logical record exceeding this size is considered an error and data will be lost. Default: 102.
TRNSTAT	Keyword defining statistic logging options for transaction level (automated tool) code of the CPT routines. ◆ TERM – Specifies that a statistic message be generated upon termination of a connection. This message contains transaction data transfer information.
TRNTRAC	Keyword defining tracing options that may be useful in debugging. This defines tracing options for the transaction level (automated tool) code of the CPT routines. ◆ ARG – Specifies that a hex dump of all CPT arguments be generated before and after every CPT service request. ◆ ENTER – Specifies that a message be generated upon entry into a transaction or automated tool program. ◆ TERM – Specifies that a message be generated during task termination processing. ◆ WTD – Specifies that a hex dump of all data be generated before every WRITEQ TD command. ◆ RTD – Specifies that a hex dump of all data be generated after every READQ TD command. ◆ XLAT – Specifies that a hex dump of all data be generated before and after every XLAT service request. ◆ TOKEN – Specifies that a hex dump of the connection token be generated upon establishment of a connection.

T09MSEND

The T09MSEND macro defines the queues that are processed by the CICS/Tools. User programs place data in the CICS transient data queues referenced in the T09MSEND macros. Data is processed from the queues according to the TRIGLEV= and TRANSID= parameters in the DCT. Define the queues with TRIGLEV parameters so CICS will run the transactions. The T09TSEND transaction (IPTs) scans T09CONFIG for a matching entry for the specified transient data queue. Data is read from the queue and a connection is built to the specified IP host and port number. The data from the queue is sent over this connection as specified in the OPTIONS keyword.

```
T09MSEND QNAME=name
        [, PORT=(number)]
        [, IPNAME=a.b.c.d]
        [, SERVICE=(name)]
        [, OPTIONS=( [IPNAME]
                     [, TRANS|RAW]
                     [, LL|SEP|ALL|FILE] ) ]
        [, SEP=(CRLF|CR|LF|X'xx..')]
        [, TRANSTBL=tblname]
        [, SNDBUF=(number,size)]
        [, RCVBUF=(number,size)]
        [, APISTAT=( [CONN] [, TERM] ) ]
        [, TRNSTAT=( [TERM] ) ]
        [, APITRAC=( [ENTER] [, ARG] [, RECV] [, SEND] [, TERM] [, PASS]
                    [, CLOSE] [, TPLERR] [, TOKEN] [, TPL] [, RLSE] [, STOR] )
        [, TRNTRAC=( [ARG] [ENTER] [, TERM] [, WTD] [, RTD] [, XLAT]
                    [, TOKEN] ) ] ]
        [, DNR=YES:NO]
```

PARAMETER	DEFINITION
QNAME	Defines the transient data queue name that contains data to be sent on a TCP/IP connection. Note: This keyword is required. ◆ <i>name</i> – Specifies the queue name to receive data from.
PORT	Defines the server well-known port number. ◆ <i>number</i> – Defines the server transport provider port number. This field, service name field, or dynamic resolution are required. Note: PORT and SERVICE are mutually exclusive.
IPNAME	Defines the host name of the server. It can be specified as an internet domain name or as an internet address in dotted decimal (see example below). ◆ <i>a.b.c.d</i> – Specifies the domain name of the destination IP host. It can be specified as an internet domain name, as in NIC.DDN.MIL, or as an internet address in dotted decimal form, such as 192.112.36.5.

PARAMETER	DEFINITION
SERVICE	Defines the transport provider service name which represent the server transport provider port number. ◆ <i>name</i> – Defines the transport provider service name character string. This field, port field or dynamic resolution are required. Note: PORT and SERVICE are mutually exclusive.
OPTIONS	Defines how data from a transient data queue is to be processed before it is sent on a TCP/IP connection. ◆ IPNAME – Specifies the first record in the transient data queue is to be interpreted as the IPNAME for the connection. Optionally, this record may contain a port number. The first few bytes of the data, up to a comma, space, or CRLF, is to be interpreted as a domain name to send the data to. If a comma appears after the domain name, the next few bytes, up to a space or CRLF, must be numeric, and will be interpreted as a port number to send the data to and the PORT= keyword will be ignored. This record will not be sent on the TCP connection. ◆ TRANS or RAW – Specifies whether (TRANS) or not (RAW) to translate the data from EBCDIC to ASCII before transmitting to remote system. ◆ LL – Specifies that a 2-byte binary length field will be placed before the data read from the CICS transient data queue and sent. The length will not include the LL field. ◆ SEP – Specifies that a separator sequence will be placed after a record is read from the CICS transient data queue. ◆ ALL – Specifies that the entire contents of the CICS transient data queue will be read and sent on a single TCP session with no indication of record boundaries. ◆ FILE – Specifies that the data read from the CICS transient data queue will be sent as one TCP session and then closed. The next READQ will be sent on a separate TCP session.
SEP	Keyword that only applies if OPTIONS=SEP. It defines the character(s) to be appended to the end of a transient data queue record. ◆ CRLF – Specifies that records are separated with the ASCII carriage return-line feed character. The ASCII character for CRLF is x'0D0A'. ◆ CR – Specifies that records are separated with the ASCII carriage return character. The ASCII character for CR is x'0D'. ◆ LF – Specifies that records are separated with the ASCII line feed character. The ASCII character for LF is x'0A'. ◆ X'XX. . ' – Specifies that records are separated with a specified hexadecimal string. The string must be 2 or 4 hexadecimal digits long. Data is sent from the TCP connection and a separator sequence is placed on the end of the record. The maximum length of one record is 32767. The terminator sequence is not translated.
TRANSTBL	Keyword defining an alternate translation table to use when processing data on connections to this port, otherwise, the default translation table specified on the T09MCICS statement is used. ◆ <i>tblname</i> – Specifies the name of the ASCII to EBCDIC translation table for this entry. It overrides the default specified in the T09MCICS statement.

200801-024030-200100

PARAMETER	DEFINITION
SNDBUF	Defines buffering space for the CPT routines. This buffer space is used to transmit data to TCP/IP. The buffer number and size are multiplied to determine total send storage allocation. Note: The combination of these must not exceed 61K. ◆ <i>number</i> – Specifies the number of output buffers to be allocated when data is to be transmitted. Note: This number should be at least 2 for performance reasons. The default is 4. ◆ <i>size</i> – Specifies the maximum size of an output or send buffer. The default is 1024.
RCVBUF	Defines buffering space for the CPT API routines. This buffer space is used to retrieve data from TCP/IP. The buffer number and size are multiplied to determine total receive storage allocation. This field is not used by the SEND tool. ◆ <i>number</i> – Specifies the number of input buffers to be allocated when data is to be received. Note: This number should be at least 2 for performance reasons. The default is 4. ◆ <i>size</i> – Specifies the maximum size of an input or receive buffer. Note: The combination of these must not exceed 61K. The default is 1024.
APISTAT	Defines statistic logging options for the API level code of the CPT routines. ◆ CONN – Specifies that a message be generated upon establishment of either a server or client connection. This message contains protocol address information. ◆ TERM – Specifies that a statistic message be generated upon termination of a connection. This message contains TRUE data transfer information.
TRNSTAT	Defines statistic logging options for transaction level (automated tool) code of the CPT routines. ◆ TERM – Specifies that a statistic message be generated upon termination of a connection. This message contains transaction data transfer information.

20124030-200100

PARAMETER	DEFINITION
APITRAC	Keyword defining tracing options that may be useful in debugging. This defines tracing options for the API level code of the CPT routines. ◆ ENTER – Specifies that a message be generated upon entry into a CPT TRUE. ◆ ARG – Specifies that a hex dump of the caller's arguments be generated upon entry and exit of a CPT TRUE. ◆ RECV – Specifies that a hex dump of the transport provider's input (RECEIVE) data be generated. ◆ SEND – Specifies that a hex dump of the transport provider's output (SEND) data be generated. ◆ TERM – Specifies that a message be generated upon exit from a CPT TRUE. ◆ PASS – Specifies that a hex dump be generated of all resources related to a passed (GIVE/TAKE) connection. ◆ CLOSE – Specifies that a hex dump be generated of all resources related to a closed connection. ◆ TPLERR – Specifies that a hex dump be generated of all transport provider parameter list requests that are completed in error. ◆ TOKEN – Specifies that a hex dump of the caller's token be generated upon entry into a CPT TRUE. ◆ TPL – Specifies that a hex dump be generated of all transport provider parameter list requests that are issued. ◆ RLSE – Specifies that a message be generated upon reception of a transport provider release indication. ◆ STOR – Specifies that a hex dump of the argument used for storage management be generated upon entry and exit from the CPT TRUE storage management routine.
TRNTRAC	Defines tracing options that may be useful in debugging. This defines tracing options for the transaction level (automated tool) code of the CTP routines. ◆ ARG – Specifies that a hex dump of all CPT arguments be generated before and after every CPT service request. ◆ ENTER – Specifies that a message be generated upon entry into a transaction or automated tool program. ◆ TERM – Specifies that a message be generated during task termination processing ◆ WTD – Specifies that a hex dump of all data be generated before every WRITEQ TD command. ◆ RTD – Specifies that a hex dump of all data be generated after every READQ TD command. ◆ XLAT – Specifies that a hex dump of all data be generated before and after every XLAT service request. ◆ TOKEN – Specifies that a hex dump of the connection token be generated upon establishment of a connection.
DNR	Indicates whether to resolve remote IP addresses into IP names with DNR calls.

200801-024030-200100

T09MEND

The T09MEND macro terminates lists of configuration parameters.



Note: This macro has no parameters, but is required and must be the last macro statement in the configuration.

Appendix **B**

CPT MANUAL PAGES

This appendix contains the Unix-style “man” pages for these sample client and server programs which run on a Unix platform:

- ◆ `c11`—CICS transient data server test client
- ◆ `sv1`—CICS transient data server test server
- ◆ `UDPCLT`—UDP test client program
- ◆ `UDPSVR`—UDP test server program

Installation information for these programs is given in **Chapter Three – INSTALLATION AND CONFIGURATION**.

200801-024030-200100

20
024030-200100

Man Pages Programs

There are two sets of programs. The `cl1` and `sv1` program set uses TCP as their transport. The `udpc1t` and `udpsvr` program set uses UDP as their transport.

The program `sv1` provides the TCP server function and listens for incoming connection requests. The program `cl1` is the TCP client and initiates a connection to the server. The program `udpsvr` provides the UDP server function and waits for UDP datagrams. The program `udpc1t` is the UDP client and sends a datagram to the server. Review the following pages to familiarize yourself with the options and parameters accompanying the program calls.

There are several ways to run the programs with CPT. The default runs against loopback. Alternatively, you may use the MVS sample programs, `CL1`, `SV2`, and `SV3` to actively set up TCP connections. For example, to run the TCP server on Unix and the TCP client on MVS, modify the `CL1` program on MVS to point to your unix host and port number. Note that the default port number for the Unix server is 2002. Next, compile and link `CL1`. Then, run the server on unix by typing:

```
sv1 -p xxx
```

where `xxx` is the port number you've selected in the `CL1` program. Run the `CL1` program on MVS and verify the output is received correctly at the server.

You may also run the server on MVS. Default ports for the listener program are set differently, depending on the program language:

LANGUAGE	DEFAULT PORT
ASM	1783
SAS/C	1683
COBOL	1983
PL/1	1583

Compile, link, and execute `SV2` and `SV3` on MVS. Execute the client on Unix with the following command:

```
cl1 -p xxx hostname
```

where `xxx` is the port number that the MVS program `SV2` is listening on. Verify that characters typed at the Unix client are received at the MVS server.

CL1 (1)

NAME

`cl1` - CICS transient data server test client

SYNOPSIS

```
cl1 [ -dt ] [ -o option ] [ -s sep ] [ -f filename ]
    [ -p port ] [ -x traceval ] [ host ]
```

DESCRIPTION

`cl1` transmits the contents of the standard input or filename to the CICS transient data server at the specified host and port using the specified option.

`cl1` operates silently unless an error is encountered or the `-d` (debug) and/or `-x` (hex trace) switches are specified. The program returns 0 if successful; otherwise, it returns 1.

The source for `cl1` is `cl1.c`. The program also calls common functions in `util.c` and `xlite.c`.

OPTIONS

- `-d` Turn on debugging mode for more verbose output. In this mode, arguments are echoed, socket function completions are noted, IP and port addresses are echoed when the connection to the server is made, and statistics are displayed at the end of the transfer. The default is not to use debugging mode.
- `-t` Translate the file data from ASCII to EBCDIC. The translate table used is table "atoc" in `xlite.c`. The default is not to translate.
- `-x traceval` Generate a hexadecimal trace of input or output data. A traceval of 0 means not to trace, 1 means to trace input data received from the network, 2 means to trace output data written to filename or to standard output, and 3 means to trace both input and output data. The default is not to trace.
- `-o option` Use the specified transfer mode option. The following options are supported: `SEP`, `LL`, `FILE`, and `ALL`. The default option is `SEP`. The case in which option is specified does not matter.

`SEP` indicates that each record of the input file is to be transferred with a separator sequence appended to the end of the record; the sequence is specified as `sep`. A record is defined as that portion of the input data up to, but not including, the terminating ASCII `LF` character.

`LL` indicates that each record is to be preceded by a two-byte length field; the length specified in the length field is the actual record length and does not include the two-byte length field itself. A record is defined as above.

`FILE` indicates that the data is to be sent "as is" but limited to 32767 or less bytes.

`ALL` indicates that all of the data is to be sent "as is".

`-s sep` When transferring with the `SEP` option, use the specified `sep` sequence to terminate each record. The following separator sequences are supported: `CRLF`, `CR`, `LF` and `hexval`. The default separator sequence is `CRLF`. This switch is only valid when the `SEP` option is specified or defaulted. The case in which `sep` is specified does not matter.

`CRLF` indicates that each record is to be terminated with an ASCII `CRLF` sequence (0x0D0A).

`CR` indicates that each record is to be terminated with an ASCII `CR` character (0x0D).

`LF` indicates that each record is to be terminated with an ASCII `LF` character (0x0A).

`hexval` indicates that the specified hexadecimal string is to be appended to the end of each record. `hexval` must specify an even number of hex digits and must begin with the sequence "0x" or "0X".

`-f filename` Read the input from the specified filename instead of from the standard input.

`-p port` Connect to the specified port. The default port is 9 (discard).

`host` Connect to the CICS server at host. The default host is "127.0.0.1" (loopback).

SV1(1)

NAME

sv1 - CICS transient data server test server

SYNOPSIS

```
sv1 [ -dt ] [ -o option ] [ -s sep ] [ -f filename ]
    [ -p port ] [ -x traceval ]
```

DESCRIPTION

sv1 listens on port and, when connected to, receives data from a CICS transient data client into filename or to the standard output using the specified option.

sv1 operates silently unless an error is encountered or the -d (debug) and/or -x (hex trace) switches are specified. The program returns 0 if successful; otherwise, it returns 1.

The source for sv1 is sv1.c. The program also calls common functions in util.c and xlate.c.

OPTIONS

- | | |
|-------------|---|
| -d | Turn on debugging mode for more verbose output. In this mode, arguments are echoed, socket function completions are noted, IP and port addresses are echoed when the connection to the server is made, and statistics are displayed at the end of the transfer. The default is not to use debugging mode. |
| -t | Translate the network data from EBCDIC to ASCII. The translate table used is table "etoa" in xlate.c. The default is not to translate. |
| -x traceval | Generate a hexadecimal trace of received input data or written output data. A traceval of 0 means to generate no trace, 1 means to trace input data received from the network, 2 means to trace output data written to filename or standard output, and 3 means to trace both input and output data. The default is not to trace any input or output data. |
| -o option | <p>Use the specified transfer mode option. The following options are supported: SEP, LL, FILE, and ALL. The default option is SEP. The case in which option is specified does not matter.</p> <p>SEP indicates that each network will be terminated by a separator sequence which will be appended to the end of the record; the sequence is specified as sep. Each record will have the separator sequence deleted and a LINEFEED appended before being written to the local file.</p> <p>LL indicates that each network record will be preceded by a two-byte length field; the length specified in the length field is the actual record length and does not include the two-byte length field itself. Each record will have the LL field stripped and a LINEFEED appended before being written to the local file.</p> |

200 24030-200100

FILE indicates that the data is to be received “as is” but limited to 32767 or less bytes (since supposedly the record is one CICS transient data record and such records are limited to a maximum of 32767 bytes). The network data will be written to the local file without modification.

ALL indicates that all of the network data is to be received “as is”. The network data will be written to the local file without modification.

-s sep

When receiving network data with the **SEP** option, use the specified **sep** sequence to terminate each record. The following separator sequences are supported: **CRLF**, **CR**, **LF**, and **hexval**. The default separator sequence is **CRLF**. This switch is only valid when **SEP** option is specified or defaulted. The case in which **sep** is specified does not matter.

CRLF indicates that each record will be terminated with an ASCII **CRLF** sequence (0x0D0A).

CR indicates that each record will be terminated with an ASCII **CR** character (0x0D).

LF indicates that each record will be terminated with an ASCII **LF** character (0x0A).

hexval indicates that the specified hexadecimal string is to be appended to the end of each record. **hexval** must specify an even number of hex digits and must begin with the sequence “0x” or “0X”.

-f filename

Write the received data to the specified filename instead of to the standard output.

-p port

Listen on the specified port. The default port is 2002.

UDPCLT (1)

NAME

udpclt - UDP test client program.

SYNOPSIS

```
udpclt [ _h_o_s_t_n_a_m_e ] [ _p_o_r_t ]
```

DESCRIPTION

udpclt prompts the user for a message, then sends the datagram via UDP to the specified _p_o_r_t on _h_o_s_t_n_a_m_e and receives the message back from the transient data server. The server also returns the length of the returned message. (See also udpsvr.)

The source for udpclt is udpclt.c.

OPTIONS

-hostname	Destination host. The default is the local hostname.
-port	The port number on which the UDP transient server is awaiting datagrams. The default port number is 12345.

UDPSVR (1)

NAME

`udpsvr` - UDP test server program.

SYNOPSIS

`udpsvr [ port ]`

DESCRIPTION

`udpsvr` waits on the specified `_p_o_r_t` and, when a datagram is received, echoes it back to the client. The length of the message is also returned. (See also `udpcvt`.)

The source for `udpsvr` is `udpsvr.c`.

OPTIONS

`-port` The port number on which the server awaits datagrams. The port number must be a positive integer. The default port number is 12345.



Appendix **C**

MESSAGES AND CODES

This appendix describes the messages and codes used by the CICS Programmer's Toolkit. It includes these sections:

- ◆ Return Code Cross Reference Table
 - ◆ Messages
 - ◆ Abend Codes
-

200801-024030-200100

200
24030-200100

Return Code Cross Reference Table

DECIMAL	HEX	VARIABLE	DESCRIPTION
0	0	CPTIRCOK	Request completed successfully
4	1	CPTWNEGO	Buffer, queue sizes reset to system limits
15	0F	CPTWEXCP	Other warning
17	11	CPTEVERS	Control block version number not supported
18	12	CPTECONN	Req host/service/port connection not found
19	13	CPTEPROT	Specified protocol not supported
20	14	CPTETOKN	Specified data transfer token is invalid
21	15	CPTEBUFF	Buffer address and/or length invalid
22	16	CPTECHAR	Translate character set is invalid
23	17	CPTEMODE	Translate mode specification is invalid
24	18	CPTECOPT	Close mode specification is invalid
25	19	CPTETABL	Specified translate table not correct
26	41A	CPTETRID	Designated transaction ID cannot start
31	1F	CPTEFRMT	Other <i>Transport Parameter List (TPL)</i> format or specification error
33	21	CPTEPBSY	Selected port is busy with active server
34	22	CPTENAPI	SNS/API not fully available; retry
35	23	CPTENAVL	Requested facility is not available
36	24	CPTEDRAN	Environment is being drained
40	28	CPTETERM	Environment is being terminated
47	2F	CPTEENVR	Other TPL environmental condition
65	41	CPTERLSE	Orderly release of remote connection request
68	44	CPTEDISC	Remote connection not available or aborted
72	48	CPTEPRGE	Remote connection environment terminating
79	4F	CPTEINTG	Other TPL connection/data integrity error
143	8F	CPTEPROC	Procedural error
254	FE	CPTABEND	Abnormal termination
255	FF	CPTEOTHR	Other error

200801-024030-200100

Messages

This section describes the messages returned by CICS.

CPT001E

DUPLICATE T09MCICS STATEMENTS

A duplicate T09MCICS macro statement found. Only one T09MCICS is valid.

Action Required

Remove duplicate T09MCICS macro statement and reassemble the configuration.

Source Module

T09CONFIG

CPT002E

SSN KEYWORD IS REQUIRED

The default SSN keyword was modified with an incorrect value.

Action Required

Correct the SSN keyword value and reassemble the configuration.

Source Module

T09CONFIG

CPT003E

TRANSID KEYWORD FOR TERM ID REQUIRED OR KEYWORD
SPECIFIED IS INVALID *tdq*

The default TRANSID keyword was modified with an incorrect value.

Action Required

Correct the TRANSID keyword value and reassemble the configuration.

Source Module

T09CONFIG

CPT004E

STATISTIC QUEUE NAME REQUIRED OR KEYWORD
SPECIFIED IS INVALID *tdq*

The default statistic transient queue name was modified with an incorrect value.

Action Required

Correct the statistic transient queue name and reassemble the configuration.

Source Module

T09CONFIG

CPT005E

TRACE QUEUE NAME REQUIRED OR KEYWORD SPECIFIED IS
INVALID *tdq*

The default trace transient queue name was modified with an incorrect value.

Action Required

Correct the trace transient queue name and reassemble the configuration.

Source Module

T09CONFIG

CPT006E

ERROR QUEUE NAME REQUIRED OR KEYWORD SPECIFIED IS
INVALID *tdq*

The default error transient queue name was modified with an incorrect value.

Action Required

Correct the error transient queue name and reassemble the configuration.

Source Module

T09CONFIG

20124030-200100

CPT007E

Error Message Text

Various messages may be generated when an Administrator Interface option is specified in error in the `T09MCICS` macro, when the macro is generating the Configuration Table.

Action Required

Correct the option in error, and resubmit the assembly.

Source Module

`T09CONFIG`

CPT008E

MISSING T09MCICS STATEMENT

A configuration macro statement was not preceded by the `T09MCICS` macro instruction. The `T09MCICS` macro statement must be the first macro statement within the configuration.

Action Required

Place the `CPTMCICS` macro statement as the first operand and reassemble the configuration.

Source Module

`T09CONFIG`

CPT009E

INVALID MF *mf*

The macro format value *mf* specified is not valid or supported. Supported options are DSECT and CSECT.

Action Required

Correct MF keyword to DSECT or CSECT and reassemble the configuration.

Source Module

T09CONFIG

CPT011E

PORT NUMBER, SERVICE NAME OR OPTIONS=(IPNAME)
REQUIRED

CPT configuration statements required the definition of a port or service operand. This is a required operand when either the T09MLSTN or T09MSEND macro statement is used.

Action Required

Add PORT or SERVICE operand to configuration macro statement and reassemble the configuration.

Source Module

T09CONFIG

200-124030-200100

CPT012E

PORT KEYWORD INVALID SPECIFICATION *port*

The `PORT` operand specified is invalid or sublist contained more than one value.

Action Required

Verify the values specified by the `PORT` operand and reassemble the configuration.

Source Module

T09CONFIG

CPT013E

TRANSID REQUIRED OR INVALID SPECIFICATION *trnid*

The `TRANSID` specified is invalid. The `TRANSID` operand is required and must be four characters.

Action Required

Add or verify the *transid* in the `T09MLSTN` configuration macro statement and reassemble the configuration.

Source Module

T09CONFIG

CPT014E

UNRECONGNIZED OR INVALID PROTOCOL *protocol*

The protocol specification or second keyword protocol on the `PORT` operand is invalid. TCP or UDP are the only valid values that can be specified.

Action Required

Correct protocol specification on `PORT` operand and reassemble the configuration.

Source Module

T09CONFIG

CPT015E

LISTEN TRANSID REQUIRED ON T09MCICS STATEMENT

A `T09MLSTN` macro statement was entered; however, a listen transaction id was not specified in the `T09MCICS` macro statement. The listen transaction id is not required, unless a `T09MLSTN` macro statement is entered.

Action Required

Add a `LISTEN` tool transaction id to the `T09MCICS` configuration macro statement and reassemble the configuration.

Source Module

T09CONFIG

20124030-200100

CPT017E

IPNAME= AND OPTIONS=(IPNAME) MUTUALLY EXCLUSIVE

The IPNAME= keyword and OPTIONS=(IPNAME) option are mutually exclusive.

Action Required

Remove either the IPNAME= or OPTIONS=(IPNAME) value and reassemble the configuration.

Source Module

T09CONFIG

CPT018E

IPNAME= OR OPTIONS=(IPNAME) IS REQUIRED

Either the IPNAME= keyword or OPTIONS=(IPNAME) option are required.

Action Required

Add either the IPNAME= or OPTIONS=(IPNAME) value and reassemble the configuration.

Source Module

T09CONFIG

CPT020E

QNAME= AND OPTIONS=(QNAME) MUTUALLY EXCLUSIVE

The QNAME= keyword and OPTIONS=(QNAME) option are mutually exclusive.

Action Required

Remove either the QNAME= or OPTIONS=(QNAME) value and reassemble the configuration.

Source Module

T09CONFIG

CPT021E

QNAME= OR OPTIONS=(QNAME) IS REQUIRED

Either the QNAME= keyword or OPTIONS=(QNAME) option are required.

Action Required

Add either the QNAME= or OPTIONS=(QNAME) value and reassemble the configuration.

Source Module

T09CONFIG

CPT022E

**OPTIONS=TRANS IS MUTUALLY EXCLUSIVE WITH
OPTIONS=RAW**

The TRANS and RAW options are mutually exclusive.

Action Required

Remove either the TRANS or RAW option and reassemble the configuration.

Source Module

T09CONFIG

20124030-200100

CPT023E

OPTIONS=STRIP IS MUTUALLY EXCLUSIVE WITH
OPTIONS=NOSTRIP

The STRIP and NOSTRIP options are mutually exclusive.

Action Required

Remove either the STRIP= or NOSTRIP option and reassemble the configuration.

Source Module

T09CONFIG

CPT024E

OPTIONS LL, SEP, ALL, AND FILE ARE MUTUALLY
EXCLUSIVE WITH EACH OTHER

The LL, SEP, ALL and FILE options are mutually exclusive. Only one data processing option is valid.

Action Required

Remove all excess options and reassemble the configuration.

Source Module

T09CONFIG

CPT025E

UNRECOGNIZED OPTIONS KEYWORD option

The option specified is invalid. Valid options include LL, SEP, ALL, and FILE.

Action Required

Correct invalid option specified and reassemble the configuration.

Source Module

T09CONFIG

CPT026E

OPTIONS=SEP REQUIRES THE SEP= KEYWORD

The SEP option was specified; however, no separator character(s) was defined.

Action Required

Add SEP= keyword with character(s) and reassemble the configuration.

Source Module

T09CONFIG

CPT027E

UNRECOGNIZED OR INVALID SEPARATOR sep

The specified separator *sep* is invalid. Verify that the separator does not exceed two characters.

Action Required

Correct invalid separator character(s) and reassemble the configuration.

Source Module

T09CONFIG

2000-24030-200100

CPT028E

QNAME REQUIRED OR INVALID SPECIFICATION *qname*

The **QNAME** operand specified is required, invalid, or sublist contained more than one value. The **T09MSEND** macro instruction requires a **QNAME** be specified.

Action Required

Verify that the value specified by the **QNAME** operand and reassemble the configuration.

Source Module

T09CONFIG

CPT029E

INVALID TDQBUF *tdq*

The transient data queue buffer value *tdq* specified is invalid.

Action Required

Correct invalid **TDQBUF** value and reassemble the configuration.

Source Module

T09CONFIG

CPT030E

UNRECOGNIZED TRNSTAT *trnstat*

The TRNSTAT keyword *trnstat* is not a valid value. Verify the keyword with value TRNSTAT values.

Action Required

Correct the TRNSTAT keyword and reassemble the configuration.

Source Module

T09CONFIG

CPT031E

UNRECOGNIZED TRNTRAC *trntrac*

The TRNTRAC keyword *trntrac* is not a valid value. Verify the keyword with value TRNTRAC values.

Action Required

Correct the TRNTRAC keyword and reassemble the configuration.

Source Module

T09CONFIG

200
24030-200100

CPT040E

SERVICE INVALID KEYWORD SPECIFICATION *service*

The `SERVICE` name operand specified is invalid or sublist contained more than one value. `SERVICE` name has a maximum length of 36 characters.

Action Required

Verify the value specified by the `SERVICE` name operand and reassemble the configuration.

Source Module

T09CONFIG

CPT041E

INVALID SEND BUFFER NUMBER *nnnnn*

The send buffer number *nnnnn* specified as the first keyword on the `SNDBUF` operand is invalid. Verify that the value specified is numeric.

Action Required

Correct send buffer number and reassemble the configuration.

Source Module

T09CONFIG

CPT042E

INVALID SEND BUFFER SIZE *nnnnn*

The send buffer size *nnnnn* specified as the second keyword on the `SNDBUF` operand is invalid. Verify that the value specified is numeric.

Action Required

Correct send buffer size and reassemble the configuration.

Source Module

T09CONFIG

CPT043E

INVALID RECEIVE BUFFER NUMBER *nnnnn*

The receive buffer number *nnnnn* specified as the first keyword on the `RCVBUF` operand is invalid. Verify that the value specified is numeric.

Action Required

Correct receive buffer number and reassemble the configuration.

Source Module

T09CONFIG

CPT044E

INVALID RECEIVE BUFFER SIZE *nnnnn*

The receive buffer number *nnnnn* specified as the second keyword on the `RCVBUF` operand is invalid. Verify that the value specified is numeric.

Action Required

Correct receive buffer size and reassemble the configuration.

Source Module

T09CONFIG

CPT045E

UNRECOGNIZED OR INVALID IPNAME *ipname*

The IP name *ipname* specified is invalid. The maximum field length is 255.

Action Required

Correct the IP name and reassemble the configuration.

Source Module

T09CONFIG

CPT046E

UNRECOGNIZED OR INVALID TRANSID *transid*

The trans ID *transid* specified is invalid. Verify that the trans ID specified is a valid transaction ID.

Action Required

Correct the trans ID and reassemble the configuration.

Source Module

T09CONFIG

CPT047E

UNRECOGNIZED APISTAT KEYWORD *apistat*

The APISTAT keyword *apistat* is not a valid value. Verify the keyword with valid APISTAT values.

Action Required

Correct the APISTAT keyword and reassemble the configuration.

Source Module

T09CONFIG

200801-024030-200100

CPT048E

UNRECOGNIZED APITRAC KEYWORD *apitrac*

The `APITRAC` keyword *apitrac* is not a valid value. Verify the keyword with value `APITRAC` values.

Action Required

Correct the `APITRAC` keyword and reassemble the configuration.

Source Module

T09CONFIG

CPT049E

DNRSSN KEYWORD IS REQUIRED OR KEYWORD SPECIFIED IS INVALID *dnrdsn*

The `DNRSSN` operand *dnrdsn* specified is invalid or sublist contained more than one value. `DNRSSN` has a maximum length of four characters.

Action Required

Verify the value specified by the `DNRSSN` operand and reassemble the configuration.

Source Module

T09CONFIG

CPT050E

**TRANSLATION TABLE IS REQUIRED OR KEYWORD
SPECIFICATION IS INVALID *transtbl***

The `TRANSTBL` operand specified is required, invalid or sublist contained more than one value. `TRANSTBL` has a maximum length of seven characters. The `T09MCICS` macro instruction requires a `TRANSTBL` be specified.

Action Required

Verify the value specified by the `TRANSTBL` operand and reassemble the configuration.

Source Module

T09CONFIG

CPT051E

**TRANSID KEYWORD FOR LSTNID REQUIRED OR KEYWORD
SPECIFIED IS INVALID *transid***

The `TRANSID` operand *transid* specified is invalid or sublist contained more than one value.

Action Required

Verify the value specified by the `TRANSID` operand and reassemble the configuration.

Source Module

T09CONFIG

CPT052E

SECURITY EXIT SPECIFICATION IS INVALID

The `SCTYEXIT` operand must be one value with a maximum length of eight characters if specified in the `T09MCICD` configuration macro.

Action Required

Verify the value specified by the `TRANSID` operand and reassemble the configuration.

Source Module

T09CONFIG

CPT056E

DUPLICATE QNAME SPECIFICATION *qname*

The `QNAME` specified has been previously configured.

Action Required

Modify or delete duplicate `qname` specification and reassemble the configuration.

Source Module

T09CONFIG

201
24030-200100

CPT057E

IPNAME INVALID KEYWORD SPECIFICATION *ipname*

The IPNAME operand specified is invalid or sublist contained more than one value. IPNAME has a maximum length of 255 characters.

Action Required

Verify the value specified by the IPNAME operand and reassemble the configuration.

Source Module

T09CONFIG

CPT058E

SEP INVALID KEYWORD SPECIFICATION *sep*

The SEP operand specified is invalid or sublist contained more than one value.

Action Required

Verify the value specified by the SEP operand and reassemble the configuration.

Source Module

T09CONFIG

CPT059E**UDP IS NOT SUPPORTED WITH AUTOMATED RECEIVE TOOL**

A LISTEN service has been configured with PROTO=UDP and PARM=xxxx. The PARM= keyword indicates the automated RECEIVE tool and the RECEIVE tool does not support UDP.

Action Required

Specify TCP as the transmission protocol or remove PARM= keyword.

Source Module

T09CONFIG

CPT060E**TRANSTBL INVALID KEYWORD SPECIFICATION *transtbl***

The TRANSTBL operand specified is invalid or sublist contained more than one value. TRANSTBL has a maximum length of seven characters.

Action Required

Verify the value specified by the TRANSTBL operand and reassemble the configuration.

Source Module

T09CONFIG

200
24030-200100

CPT061E

SNDBUF INVALID SUBLIST SPECIFICATION *sndbuf*

The `SNDBUF` operand specified is invalid or sublist contained more than two values.

Action Required

Verify the values specified by the `SNDBUF` operand and reassemble the configuration.

Source Module

T09CONFIG

CPT062E

RCVBUF INVALID SUBLIST SPECIFICATION *sndbuf*

The `RCVBUF` operand specified is invalid or sublist contained more than two values.

Action Required

Verify the values specified by the `RCVBUF` operand and reassemble the configuration.

Source Module

T09CONFIG

CPT063E

DUPLICATE PORT SPECIFICATION *port*

The **PORT** specified has been previously configured.

Action Required

Modify or delete duplicate port specification and reassemble the configuration.

Source Module

T09CONFIG

CPT064E

DUPLICATE SERVICE SPECIFICATION *service*

The **SERVICE** name specified has been previously configured.

Action Required

Modify or delete duplicate service name specification and reassemble the configuration.

Source Module

T09CONFIG

20
24030-200100

CPT066E

TDQBUF INVALID KEYWORD SPECIFICATION *tdqbuf*

The TDQBUF operand specified is invalid or sublist contained more than one value.

Action Required

Verify the value specified by the TDQBUF operand and reassemble the configuration.

Source Module

T09CONFIG

CPT067E

TDQBUF EXCEEDED MAX OF 32,767 *tdqbuf*

The TDQBUF operand specified exceeded the maximum value allowed.

Action Required

Reduce the value specified by the TDQBUF operand and reassemble the configuration.

Source Module

T09CONFIG

CPT068E

TRANSID INVALID SUBLIST SPECIFICATION *transid*

The TRANSID operand specified is invalid or sublist contained more than two values.

Action Required

Verify the values specified by the TRANSID operand and reassemble the configuration.

Source Module

T09CONFIG

CPT069E

QNames INVALID SUBLIST SPECIFICATION *qnames*

The QNames operand specified is invalid or sublist contained more than three values.

Action Required

Verify the values specified by the QNames operand and reassemble the configuration.

Source Module

T09CONFIG

20124030-200100

CPT100I

mod_name CPT TRUE EXIT INTERFACE ENABLED

CPT initialization has successfully enabled the *Task-Related User Exit (TRUE)* interface routine. The CPT interface is not available until after the CPT109I message.

Action Required

This is a normal CPT initialization message.

Source Module

mod_name

CPT101I

mod_name CPT ESTABLISHED API SUBSYSTEM SESSION
SSN

CPT initialization has successfully established a session with the API subsystem *SSN* and is communicating with the transport provider.

Action Required

This is a normal CPT initialization message.

Source Module

mod_name

CPT102I

mod_name CPT INITIALIZATION PENDING API
SUBSYSTEM OPEN

CPT initialization was unable to open or establish a session with the API. CPT initialization will retry for 15 seconds. If subsystem still fails to open, CPT will not initialize.

Action Required

Start the SNS/TCPaccess API or transport provider.

Source Module

mod_name

CPT103I

mod_name CPT TERMINATED API SUBSYSTEM SESSION

CPT termination has successfully closed the API session. Communication with the transport provider will not be available.

Action Required

This is a normal CPT termination message.

Source Module

mod_name

20
024030-200100

CPT104E

***mod_name* API SUBSYSTEM SESSION PREVIOUSLY
ESTABLISHED**

CPT initialization has found the API session has already been established.

Action Required

Verify that CPT initialization has not run twice and contact your CICS technical support personnel.

Source Module

mod_name

CPT105E

***mod_name* CPT INITIALIZATION FAILED**

A CPT initialization routine has failed. This message indicates a serious error within the CPT interface. Review previous messages reported during system initialization for additional information.

Action Required

Previous messages should indicate the cause of failure.

Source Module

mod_name

200801-024030-200100

CPT106I

mod_name CPT API ENDPOINT *nnnnnnnn* CLOSED

CPT termination routines have closed an open CICS API endpoint or connection. *nnnnnnnn* is the connection token.

Action Required

This is a normal CPT termination message.

Source Module

mod_name

CPT107I

mod_name CPT PENDING TERMINATION ECB

CPT termination routines are waiting for a connection(s) to be closed. When complete, message CPT103I will be issued.

Action Required

This is a normal CPT termination message.

Source Module

mod_name

20124030-200100

CPT108E

mod_name CPT TERMINATION ERROR; SHUTDOWN
INCOMPLETE

A CPT termination routine has failed. This is a serious error and should be report to the system administrator.

Action Required

A previous message should indicate the cause.

Source Module

mod_name

CPT109I

mod_name CPT INITIALIZATION SUCCESSFUL

CPT initialization has successfully completed. Both the automated transactions and programmer interface are available for users.

Action Required

This is a normal CPT initialization message.

Source Module

mod_name

200801-024030-200100

CPT110E
CPT110E
CPT110E
CPT110E
CPT110E

mod_name CPT TRUE EXIT ENABLE FAILED; INTERFACE
NOT USABLE
mod_name VERIFY THE FOLLOWING:
mod_name 1. DEFINED VIA THE PPT OR VIA RDO
mod_name 2. NOT PPT DISABLED
mod_name 3. PRESENT IN A DFHRPL LOAD LIBRARY

A program necessary for CPT to successfully initialize was not usable. A CICS START, ENABLE, or LOAD command failed. *mod_name* is the module name that failed.

Action Required

Verify the steps outlined in the message. Review CPT Installation and Customization requirements.

Source Module

mod_name

CPT111E

mod_name CPT TRUE EXIT ENABLE FAILED; INTERFACE
ALREADY ENABLED

CPT initialization encountered an error code which indicates that initialization has already been run.

Action Required

Verify that the CPT successfully terminated previously or that program CPTTSTRT has not been improperly run by another program or transaction.

Source Module

mod_name

20
024030-200100

CPT111I

mod_name CPT DEFAULT TRANSLATION TABLE LOADED

The default translation table defined in the CPTMCICS macro instruction has been successfully loaded.

Action Required

This is a normal CPT initialization message.

Source Module

mod_name

CPT112E

mod_name CPT TRUE EXIT ENABLE FAILED; EXIT=YES
SPECIFIED IN THE SIT

CPT initialization encountered an error code indicating exits are not allowed.

Action Required

Change the CICS SIT to allow exits.

Source Module

mod_name

CPT112I

mod_name CPT DEFAULT TRANSLATION TABLE RELEASED

The default CPT translation table has been released.

Action Required

This is a normal CPT termination message.

Source Module

mod_name

CPT113E

mod_name CPT TRUE EXIT ENABLE FAILED; UNEXPECTED
ERROR HAS OCCURRED

CPT initialization encountered an error code that indicates that an unrecognized error has occurred.

Action Required

Determine the error code and contact Interlink Customer Support.

Source Module

mod_name

CPT114E

mod_name CPT TRUE EXIT INITIALIZATION ROUTINE
FAILED RC=xx

CPT TRUE initialization routine, CPTCINIT, failed. Return code should indicate the cause of failure.

Action Required

A previous message should indicate the error. Read the section **Return Code Cross Reference Table** on page C-3 for descriptions.

Source Module

mod_name

20
24030-200100

CPT115E

***mod_name* CPT INITIALIZATION FAILED FOR
SUBSYSTEM-*ssn***

A CPT initialization routine has failed for a particular subsystem version. This message indicates a serious error within the CPT interface. Review previous messages reported during system initialization for additional information.

Action Required

Previous messages should indicate the cause of failure.

Source Module

mod_name

CPT120I

***mod_name* CPT TERMINATION SUCCESSFUL**

The CPT interface has been shut down. CPT termination completed successfully.

Action Required

This is a normal CPT termination message.

Source Module

mod_name

CPT121E

mod_name CPT TERMINATION ROUTINE FAILED

The CPT interface was unable to terminate properly. CPTCTERM TRUE routine returned a bad return code.

Action Required

Previous messages generated by CPTCTERM should indicate the source of the error.

Source Module

mod_name

CPT122E

mod_name CPT INTERFACE NOT ACTIVE

Either CICS shutdown was initiated or the CPT shutdown transaction was issued and the CPT interface was not enabled.

Action Required

This does not indicate a specific error. Review the status of CPT Interface.

Source Module

mod_name

20124030-200100

CPT123I

mod_name INITIAL WRITE TO ERROR LOG TD QUEUE

CPT initialization has successfully verified error log transient data queue is accessible.

Action Required

This is a normal CPT initialization message.

Source Module

mod_name

CPT125I

mod_name CPT SUCCESSFULLY COMPLETED DNR REQUEST
TO *dnrdsn*

CPT initialization has successfully verified communication with the *Domain Name Resolver (DNR)*. The DNR subsystem is *dnrdsn*.

Action Required

This is a normal CPT initialization message.

Source Module

mod_name

CPT126I

mod_name CPT FAILED INITIAL DNR REQUEST TO *dnrdsn*

The initial CPT communication with the DNR failed to complete successfully. This message indicates a serious error and CPT initialization will fail. The DNR subsystem is *dnrdsn*.

Action Required

Verify the *DNRSSN* operand and reassemble the configuration.

Source Module

mod_name

CPT130E

mod_name CPT PROGRAMMER INTERFACE DISABLED

This message is part of normal shutdown processing and indicates that the TRUE interface has been disabled.

Action Required

This is a normal CPT termination message.

Source Module

mod_name

20124030-200100

CPT131E

***mod_name* CPT CONFIGURATION TABLE CPTCONFIG
RELEASED**

This message is part of normal shutdown processing and indicates that the CPTCONFIG has been released.

Action Required

This is a normal CPT termination message.

Source Module

mod_name

CPT132E

***mod_name* CPT TRUE EXIT INTERFACE DISABLED**

The TRUE interface has been disabled with the CICS `DISABLE` command.

Action Required

This is a normal CPT termination message or indicates a failure of some required activity during initialization.

Source Module

mod_name

CPT197E

mod_name TRUE EXIT CALLER WITH ODD R0

The CPT TRUE interface was entered with an invalid function code in register 0.

Action Required

Validate transaction CPT stub component for calling transaction. This may require a dump of the transaction.

Source Module

mod_name

CPT198E

mod_name TRUE EXIT CALLER WITH R0 INVALID

The CPT TRUE interface was entered with an invalid function code in register 0.

Action Required

Validate transaction CPT stub component for calling transaction. This may require a dump of the transaction.

Source Module

mod_name

200
J24030-200100

CPT199I

mod_name TASK MANAGER FUNCTION

The CPT TRUE interface was entered with the task manager function code.

Action Required

This message is only informational and is an error condition.

Source Module

mod_name

CPT200E

mod_name ENTERED WITH BAD TRUE EXIT FUNCTION CODE
nnnnn

The CPT TRUE was entered with an invalid function code. *nnnnn* is the function code.

Action Required

Determine how the exit was called and contact Interlink Customer Support.

Source Module

mod_name

CPT201E

mod_name ENTERED WITH BAD TRUE EXIT CALLER ID
nnnnn

The CPT TRUE was entered with an invalid caller ID. *nnnnn* is the caller ID.

Action Required

Determine how the exit was called and contact Interlink Customer Support.

Source Module

mod_name

CPT202E

mod_name ENTERED WITH INVALID GLOBAL AREA LENGTH
nnnnn

The CPT TRUE was entered with a global area which is too short. *nnnnn* is the length provided.

Action Required

Verify that the exit was properly enabled and contact Interlink Customer Support.

Source Module

mod_name

20124030-200100

CPT203E

mod_name CPT GLOBAL DATA AREA NOT FOUND OR NOT
INITIALIZED

The CPT TRUE was entered with the global area not initialized.

Action Required

Verify that the global area was properly initialized by CPTTSTRT and contact Interlink Customer Support.

Source Module

mod_name

CPT205E

mod_name ENTERED WITH CPT APCB NOT OPENED YET

The CPT TRUE was entered with the API APCB not open.

Action Required

Verify that the APCB has been properly built and opened by CPTTSTRT. Also, note that the APCB is closed as part of shutdown processing.

Source Module

mod_name

CPT206E

mod_name ENTERED WITH INVALID TASK WORK AREA
LENGTH *nnnnn*

The CPT TRUE was entered with task area which is too short. *nnnnn* is the length provided.

Action Required

Verify that the exit was properly enabled and contact Interlink Customer Support.

Source Module

mod_name

CPT207E

CPT ERROR ROUTINE CALLED WITH BAD MESSAGE NUMBER
nnnnn

nnnnn is the message number. This message should not occur.

Action Required

Correct the message number or call Interlink Customer Support.

Source Module

CPTCERR

206-24030-200100

CPT208E

**CPT STATS ROUTINE CALLED WITH BAD MESSAGE NUMBER
*nnnnn***

nnnnn is the message number. This message should not occur.

Action Required

Correct the message number or call Interlink Customer Support.

Source Module

CPTCSTAT

CPT209E

***mod_name* CPT CONNECTION MANAGEMENT ROUTINE WAS
UNABLE TO DETERMINE IP ADDRESS OR HOST NAME**

The API `DIRSRV` routine was unable to convert the supplied IP name to an IP address.

Action Required

Correct the name, correct the name server, or code the IP address instead of the IP name.

Source Module

mod_name

CPT210E

mod_name CPT CONNECTION MANAGEMENT ROUTINE WAS
UNABLE TO DETERMINE PORT

The API DIRSRV routine was unable to convert the supplied IP service name to an IP port number.

Action Required

Correct the name, correct the name server, or code the IP port number instead of the IP service name.

Source Module

mod_name

CPT211E

mod_name CPT CONNECTION MANAGEMENT REQUEST
FAILED DUE TO AN INVALID PROTOCOL NUMBER

The CPT interface only supports TCP and UDP. The protocol number for TCP is 6. The protocol number for UDP is 17.

Action Required

Supply the correct protocol number.

Source Module

mod_name

200
24030-200100

CPT212E

mod_name CPT CONNECTION MANAGEMENT REQUEST DID
NOT CONTAIN PORT OR SERVICE SPECIFICATION

A CPT routine required port or service information and its parameter list had none.

Action Required

Correct the parameter list or contact Interlink Customer Support.

Source Module

mod_name

CPT213E

mod_name ARGUMENT CONTAINED INVALID TOKEN. HEX
DUMP OF ARGUMENT FOLLOWS

A CPT service request was issued with an invalid token. Verify token in the argument list.

Action Required

CPT connection management trace `ARG` and `TOKEN` options can be specified to help debug bad token.

Source Module

mod_name

200801-024030-200100

CPT216E

mod_name ARGUMENT CONTAINED INVALID VERSION
NUMBER; HEX DUMP OF ARGUMENT FOLLOWS

A CICS/API service was called with an invalid version number. Currently, only binary 1 is a valid version number.

Action Required

Correct version number set by calling program.

Source Module

mod_name

CPT218E

mod_name TOKEN DID NOT CONTAIN API ENDPOINT ID.
HEX DUMP OF TOKEN FOLLOWS

A CPT service request was issued with an invalid token or corrupted token.
Verify token in the argument list.

Action Required

CPT connection management trace `ARG` and `TOKEN` options can be specified to help debug bad token.

Source Module

mod_name

CPT219E

API TRACE ROUTINE CALLED WITH BAD MESSAGE NUMBER
nnnnn

nnnnn is the message number. This message should not occur.

Action Required

Correct the message number or call Interlink Customer Support.

Source Module

CPTCTRAC

CPT226E

mod_name CPT LISTEN ROUTINE FAILED, RC=*rrrrrrrr*
DC=*ddddddd*

The CPT LISTEN interface routine failed. The return code is *rrrrrrrr*. The diagnostic code is *ddddddd*, which may be a *Transport Parameter List (TPL)* return code field.

Action Required

Refer to the CPT service return code information and the API manual for an explanation of diagnostic code.

Source Module

mod_name

CPT227E

mod_name CPT CONNECT ROUTINE FAILED, RC=rrrrrrrr
DC=ddddddd

The CPT `CONNECT` interface routine failed. The return code is *rrrrrrrr*. The diagnostic code is *ddddddd*, which may be a TPL return code field.

Action Required

Refer to the CPT service return code information and the API manual for an explanation of diagnostic code.

Source Module

mod_name

CPT228E

mod_name CPT RECV ROUTINE FAILED, RC=rrrrrrrr
DC=ddddddd

The CPT `RECEIVE` interface routine failed. The return code is *rrrrrrrr*. The diagnostic code is *ddddddd*, which may be a TPL return code field.

Action Required

Refer to the CPT service return code information and the API manual for an explanation of diagnostic code.

Source Module

mod_name

CPT229E

mod_name CPT SEND ROUTINE FAILED, RC=rrrrrrrr
DC=ddddddd

The CPT SEND interface routine failed. The return code is rrrrrrrr. The diagnostic code is ddddddd, which may be a TPL return code field.

Action Required

Refer to the CPT service return code information and the API manual for an explanation of diagnostic code.

Source Module

mod_name

CPT230E

mod_name CPT CLOSE ROUTINE FAILED, RC=rrrrrrrr
DC=ddddddd

The CPT CLOSE routine failed. The return code is rrrrrrrr. The diagnostic code is ddddddd, which may be a TPL return code field.

Action Required

Refer to the CPT service return code information and the API manual for an explanation of diagnostic code.

Source Module

mod_name

CPT231E

mod_name TADDR FAILED; HEX DUMP OF TPL FOLLOWS

An API TADDR function failed.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT232E

mod_name TINFO FAILED; HEX DUMP OF TPL FOLLOWS

An API TINFO function failed.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT233E

mod_name TCONNECT FAILED; HEX DUMP OF TPL FOLLOWS

An API TCONNECT function failed.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

20124030-200100

CPT234E

mod_name TCONFIRM FAILED, DC=dddddddd PROTOCOL
ADDR=1234567890123456

An API TCONFIRM function failed. The diagnostic code is dddddddd. The
PROTOCOL ADDR is 1234567890123456.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT235E

mod_name TOPEN FAILED; HEX DUMP OF TPL FOLLOWS

An API TOPEN function failed.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT236E

***mod_name* BIND FAILED, DC=ddddddd PROTOCOL
ADDR=dddppppaaaaaaa**

An API `TBIND` function failed. The diagnostic code is `ddddddd`. The `PROTOCOL` `ADDR` indicates the domain (`ddddddd`), port (`ddddddd`), and IP host address (`ddddddd`) in hex. Generally, this error is seen with a return code `x'21'` (decimal 33), and indicates the request port is busy. Only one Listener task can be using a port at any one time, including Listener tasks in another CICS address space.

Action Required

Refer to the API manual to interpret the Diagnostic Code. Use the Administrator Interface to check if another CICS task is using that port in the same CICS system or in another CICS system.

This error will occur if two CICS systems try to use the same Configuration Table module. The Listen tools defined for specific ports will fail when the CPT Interface is initialized in the second CICS system. In this case, you must create a separate Configuration Table source for each CICS system, and assemble the `T09CONFIG` module into a load library unique to that CICS system.

Source Module

mod_name

CPT237E

***mod_name* TOPTION FAILED; HEX DUMP OF TPL FOLLOWS**

An API `TOPTION` function failed.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT238E

mod_name TREJECT FAILED; HEX DUMP OF TPL FOLLOWS

An API TREJECT function failed.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT239E

mod_name TLISTEN FAILED; HEX DUMP OF TPL FOLLOWS

An API TLISTEN function failed.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT240E

mod_name TACCEPT FAILED; HEX DUMP OF TPL FOLLOWS

An API TACCEPT function failed.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

200801-024030-200100

CPT241E

mod_name DEQ LOGIC ERROR IN *nnnnnnnn* QUEUE

A CPT routine attempted to take a work element off queue *nnnnnnnn* and the request failed.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

CPT242E

mod_name TRETRACT FAILED; HEX DUMP OF TPL FOLLOWS

An API TRETRACT function failed.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

200-024030-200100

CPT243E

mod_name UNKNOWN CLOSE TYPE *nnnnn*

The CPT `CLOSE` interface routine has been called with an invalid value in the close type field.

Action Required

Correct the close type field.

Source Module

mod_name

CPT244E

mod_name TRELEASE FAILED; HEX DUMP OF TPL FOLLOWS

An API error has occurred in a CPT routine. The `TRELEASE` TPL is dumped.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT245E

mod_name TRELACK FAILED; HEX DUMP OF TPL FOLLOWS

An API error has occurred in a CPT routine. The `TRELACK` TPL is dumped.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT246E

mod_name TDISCONN FAILED; HEX DUMP OF TPL FOLLOWS

An API error has occurred in a CPT routine. The TDISCONN TPL is dumped.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT247E

mod_name TUNBIND FAILED; HEX DUMP OF TPL FOLLOWS

An API error has occurred in a CPT routine. The TUNBIND TPL is dumped.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT248E

mod_name TCLOSE FAILED; HEX DUMP OF TPL FOLLOWS

An API error has occurred in a CPT routine. The TCLOSE TPL is dumped.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

20
024030-200100

CPT249E

mod_name TSEND FAILED, DC=dddddddd
DALEN=12345678

An API error has occurred in a CPT routine. The diagnostic code is ddddddd. The DALEN is 12345678.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT250E

mod_name UNCHAIN FAILED FOR GLOBAL CHAIN

The global chain of active sessions did not contain the session being closed.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

CPT251E

mod_name UNCHAIN FAILED FOR TASK CHAIN

The task chain of active sessions did not contain the session being closed.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

200801-024030-200100

CPT252E

mod_name API TOPEN FAILED DUE TO component NOT AVAILABLE

An API TOPEN function was executed and returned in error. The error code indicates that a component within the transmission path was not available.

Action Required

This message will appear when the CPT application is enabled before the API interface has become available. The internal transaction will retry TOPEN function until successful.

Source Module

mod_name

CPT253E

mod_name TOKEN TO BE TAKEN (PASS) NOT FOUND, BAD TOKEN BELOW

A transaction has requested a token to be taken from one task to another. However, the token to be processed is not on the active token queue. The request is failed with the first 16 bytes of the token below this message.

Action Required

Verify token as valid CPT token or contact Interlink Customer Support.

Source Module

mod_name

20
124030-200100

CPT254E

mod_name TOKEN TO BE TAKEN (PASS) NOT IN GIVE
STATE OR ALREADY TAKEN, BAD TOKEN BELOW

A transaction has requested a token to be taken from one task to another. However, the token to be processed is not in a valid state to be passed. The request is failed with a dump of the token.

Action Required

Verify token as valid CPT token or contact Interlink Customer Support.

Source Module

mod_name

CPT256E

mod_name TPL DOES NOT CONTAIN VALID EPID

An transport provider API request was issued. However, no connection (Entry Point ID) with the API existed. A hex dump of the API TPL is provided.

Action Required

This indicates a serious logic error. Tracing options should be enabled and transaction rerun for additional diagnostic information

Source Module

mod_name

200801-024030-200100

CPT257E

mod_name GIVE (PASS) SERVICE FAILED, RC=rrrrrrrrr
DC=ddddddddd

The CPT `GIVE` interface routine failed. The return code is *rrrrrrrrr*. The diagnostic code is *ddddddddd*, which is generally not used by the `GIVE` service. Both return codes are hexadecimal values.

Action Required

Refer to the CPT service return code information for an explanation.

Source Module

mod_name

CPT258E

mod_name TAKE (PASS) SERVICE FAILED, RC=rrrrrrrrr
DC=ddddddddd

The CPT `TAKE` interface routine failed. The return code is *rrrrrrrrr*. The diagnostic code is *ddddddddd*, which is generally not used by the `TAKE` service. The `TAKE` service can be issued implicitly through out services. Both return codes are hexadecimal values.

Action Required

Refer to the CPT service return code information for an explanation.

Source Module

mod_name

200-24030-200100

CPT280E

mod_name INVALID DATA TRANSFER ARGUMENT BUFFER ADDRESS. HEX DUMP OF DATA TRANSFER ARGUMENT FOLLOWS

The ADTBUFFA or adt_buffa in the *Argument for Data Transfer (ADT)* is invalid.

Action Required

Correct the SEND argument or contact Interlink Customer Support.

Source Module

mod_name

CPT281E

mod_name INVALID DATA TRANSFER ARGUMENT LENGTH FIELD. HEX DUMP OF DATA TRANSFER ARGUMENT FOLLOWS

The ADTBUFFL or adt_buffl in the ADT is invalid.

Action Required

Correct the RECV request or contact Interlink Customer Support.

Source Module

mod_name

CPT282E

mod_name API RECEIVE TCHECK PROCESSING ACTIVE
TRECQ Q CONTAINS ZERO ENTRIES

A CPT routine tried to process its TRECQ queue for TCHECK and found no entries.
This is a logic error.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

CPT283E

mod_name API RECEIVE WAIT PROCESSING ACTIVE TRECQ
Q CONTAINS ZERO ENTRIES

A CPT routine tried to process its TRECQ queue for WAITS and found no entries.
This is a logic error.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

200
24030-200100

CPT301E

T09CADIN CPT STATISTICS CAPTURE

This message is generated at CPT shutdown or when statistics are reset through the Administrator Interface if a discard queue was specified in the Configuration Table, but that queue is either not defined to CICS or is not enabled.

Action Required

Verify the queue name and its availability.

Source Module

T09CADIN

CPT400E

***mod_name* API TPL ERROR; HEX DUMP OF TPL FOLLOWS**

An API error has occurred in a CPT routine. The TPL is dumped.

Action Required

Refer to the API manual to interpret the TPL.

Source Module

mod_name

CPT405E

mod_name DIRSRV REQUEST FAILED, RC=*rrrrrrrr*
DC=*dddddddd*

An API DIRSRV request failed. The return code is *rrrrrrrr*. The diagnostic code is *dddddddd*. This message is followed by the CPT406E and CPT407E messages.

Action Required

Read **Return Code Cross Reference Table** on page C-3 and the API manual for an explanation of diagnostic code.

Source Module

mod_name

CPT406E

mod_name DIRSRV REQUEST NAME FIELD

A CPT routine used directory services and is dumping the name field of the request.

Action Required

This information is used by the DNR service request and can be used to debug DNR service request errors.

Source Module

mod_name

200-24030-200100

CPT407E

***mod_name* DIRSRV REQUEST VALUE FIELD**

A CPT routine used directory services and is dumping the value field of the request.

Action Required

This information is used by the DNR service request and can be used to help debug DNR service request errors.

Source Module

mod_name

CPT410E

***mod_name* API SYSTEM DRAIN NOTIFICATION**

The CPT routines have been notified that the API is in drain shutdown.

Action Required

Established CPT connections will continue, but new connections will fail. Listening requests will be terminated. This is a normal condition when the API is being shut down.

Source Module

mod_name

CPT411E

***mod_name* API SYSTEM STOP NOTIFICATION**

The CPT routines have been notified that the API is in stop shutdown.

Action Required

CPT sessions will stop. This is a normal condition when the API is being shut down.

Source Module

mod_name

CPT412E

***mod_name* API SYSTEM TERMINATED**

The CPT routines have been notified that the API is in terminate shutdown.

Action Required

CPT sessions will terminate. This is a normal condition when the API is being shut down.

Source Module

mod_name

CPT413E

***mod_name* DISCONNECT INDICATION RECEIVED**

A CPT session has received a disconnect indication.

Action Required

This indicates the remote system sent a TCP RESET.

Source Module

mod_name

20124030-200100

CPT414E

mod_name RELEASE INDICATION RECEIVED

A CPT session has received a release indication.

Action Required

This indicates the remote system sent a TCP FIN. This is a normal session termination event.

Source Module

mod_name

CPT415E

mod_name LISTEN PURGED BY TCLOSE

A CPT listen request was purged because the connection was closed.

Action Required

This is a normal CPT termination message for established connections.

Source Module

mod_name

CPT420E

mod_name ABENDED

A CPT routine has abended. This is a serious error.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

CPT451E

mod_name STAT ROUTINE FAILED

A CPT routine has been unable to log its statistics information.

Action Required

Other messages should indicate the reason for this failure.

Source Module

mod_name

CPT452E

mod_name TRACE ROUTINE FAILED

A CPT routine has been unable to log its trace information.

Action Required

Other messages should indicated the reason for this failure.

Source Module

mod_name

CPT453E

mod_name FLUSHING DATA BEFORE TRELACK

A CPT session has been closed, but the remote host is still sending data. This data is being flushed.

Action Required

Determine if the application design allows for this situation to happen.

Source Module

mod_name

200-24030-200100

CPT500E

mod_name CICS EIBFN=*func_code* HANDLE CONDITION
EIBRESP=*cond_code*

An unknown CICS command failed. The CICS command function code is *func_code* and the condition code is *cond_code*.

Action Required

The EIBFN and EIBRESP values are described in the **CICS Application Programmer's Reference**. This message should be reported to the CPT administrator.

Source Module

mod_name

CPT501E

mod_name CICS GETMAIN HANDLE CONDITION *type*

The CICS GETMAIN command failed with the *type* condition.

Action Required

Verify storage is available within the CICS region. This may be a temporary error.

Source Module

mod_name

CPT502E

mod_name CICS FREEMAIN HANDLE CONDITION *type*

The CICS FREEMAIN command failed with the type condition.

Action Required

Verify storage area is owned or has been obtained by the task. This may require trace options to be specified to debug condition.

Source Module

mod_name

CPT503E

mod_name CICS LOAD HANDLE CONDITION *type* FOR *name*

The CICS LOAD command for name failed with the type condition.

Action Required

Check condition and verify the module name exists in the DFHRPL library list.

Source Module

mod_name

CPT504E

mod_name CICS START HANDLE CONDITION *type* FOR *name*

The CICS START command for name failed with the type condition.

Action Required

Check condition and verify trans id name in the CICS PCT.

Source Module

mod_name

20124030-200100

CPT505E

mod_name CICS WAIT EVENT HANDLE CONDITION *type*

The CICS WAIT command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT506E

mod_name CICS RETRIEVE HANDLE CONDITION *type*

The CICS RETRIEVE command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT507E

mod_name CICS WRITEQ TD HANDLE CONDITION *type* FOR
name

The CICS WRITEQ TD command for name failed with the type condition.

Action Required

Check condition and verify name in the CICS DCT.

Source Module

mod_name

CPT508E

mod_name CICS READQ TD HANDLE CONDITION *type* FOR
name

The CICS READQ TD command for name failed with the type condition.

Action Required

Check condition and verify name in the CICS DCT.

Source Module

mod_name

CPT509E

mod_name CICS ASSIGN HANDLE CONDITION *type*

The CICS ASSIGN command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT510E

mod_name CICS SYNCPOINT HANDLE CONDITION *type*

The CICS SYNCPOINT command for name failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT511E

mod_name CICS PUSH HANDLE CONDITION *type*

The CICS PUSH command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT512E

mod_name CICS POP HANDLE CONDITION *type*

The CICS POP command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT513E

mod_name CICS DELETEDQ HANDLE CONDITION *type* FOR
name

The CICS DELETEDQ TD command for name failed with the type condition.

Action Required

Check condition and verify name in the CICS DCT.

Source Module

mod_name

CPT514E

mod_name CICS RETURN HANDLE CONDITION *type*

The CICS RETURN command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT515E

mod_name CICS RELEASE HANDLE CONDITION *type* FOR
name

The CICS RELEASE command for name failed with the type condition.

Action Required

Check condition and verify name in CICS PPT.

Source Module

mod_name

CPT516E

mod_name CICS ABEND HANDLE CONDITION *type*

The CICS ABEND command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

20124030-200100

CPT517E

mod_name CICS HANDLE CONDITION *type*

The CICS HANDLE command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT518E

mod_name CICS ASKTIME HANDLE CONDITION *type*

The CICS ASKTIME command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT519E

mod_name CICS DELAY HANDLE CONDITION *type*

The CICS DELAY command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT520E

mod_name CICS ENTER HANDLE CONDITION *type*

The CICS ENTER command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT521E

mod_name CICS DUMP HANDLE CONDITION *type*

The CICS DUMP command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT522E

mod_name CICS RECEIVE HANDLE CONDITION *type*

The CICS RECEIVE command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT523E

mod_name CICS SEND HANDLE CONDITION *type*

The CICS SEND command failed with the type condition.

Action Required

Check condition and report error to site administrator.

Source Module

mod_name

CPT524I

mod_name BUFFER SPECIFICATION NEGOTIATED
DOWNWARD TO SYSTEM LIMITS. HEX DUMP OF NEW VALUES
FOLLOW:

The value(s) specified in the configuration macro instruction for API buffering exceeded the transport provider's maximums allowed.

Action Required

This is a warning message generated by the CPT tools transactions. The values specified in the CPTMLSTN or CPTMSEND configuration macro instruction for RCVBUF or SNDBUF should be verified and reduced.

Source Module

mod_name

CPT525E

ERROR	LOGGING TERMINATED DUE TO NOSPACE
TRACE	CONDITION
STATS	

This message is generated when the NOSPACE condition is raised while writing messages to the named transient Data Queue. For TRACE and STATS, the message is written to the ERROR Queue. If NOSPACE occurs while writing to the ERROR Queue, then the message is issued as a WTO. The function will not be attempted again until CPT is restarted.

Action Required

Verify that enough space is allocated to handle the maximum number of messages which may be generated during the lifetime of a CPT cycle.

Source Module

T09CERR, T09CTRAC, T09CSTAT

CPT600E

***mod_name* DESTINATION IP NAME NOT FOUND IN FIRST
RECORD FOR TRANSIENT QUEUE *name***

The SEND tool was not able to resolve the host name or port value from the first record read from the transient data queue. The CPTMSEND macro instruction specified dynamic server resolution with the OPTIONS= (IPNAME) keyword.

Action Required

Verify the first record on the transient data queue name for correct host name or port specification. The transient data queue in error will have to be deleted or purged to correct problem.

Source Module

mod_name

20
124030-200100

CPT601E

***mod_name* DESTINATION IP NAME IN 1ST RECORD FOUND TO BE INVALID. HEX DUMP OF RECORD FOLLOWS**

The SEND tool was not able to resolve the host name or port value from the first record read from the transient data queue. The CPTMSEND macro instruction specified dynamic server resolution with the OPTIONS= (IPNAME) keyword.

Action Required

Verify that the transaction was started properly by the CPT listen transaction and contact Interlink Customer Support.

Source Module

mod_name

CPT602E

***mod_name* TRANSACTION RETRIEVE DATA IS NOT A VALID TOKEN**

The CPT receive transaction issued a CICS RETRIEVE command which returned an invalid token.

Action Required

Verify that the transaction was started properly by the CPT listen transaction and contact Interlink Customer Support.

Source Module

mod_name

200801-024030-200100

CPT603E

mod_name TRANSACTION TOKEN DID NOT CONTAIN A USER
CONTEXT FIELD

The CPT receive transaction issued a CICS `RETRIEVE` command which returned a value that was not properly formatted. The user context field was zero.

Action Required

Verify that the transaction was started properly by the CPT listen transaction and contact Interlink Customer Support.

Source Module

mod_name

CPT604E

mod_name ERROR WHILE PROCESSING QNAME

CPT automated tool received an error condition while processing a transient data queue.

Action Required

Review previous messages generated in the CPT error log to determine cause of problem.

Source Module

mod_name

200
24030-200100

CPT605E

mod_name NO TRANSLATION TABLE NAME FOUND IN
CPTCONFIG FILE

The configuration entry specified translate but the translation table was not specified.

Action Required

The configuration macros should not allow this to successfully assemble. Verify that the macros have not been modified and that the assembly was successful.

Source Module

mod_name

CPT606E

mod_name INVALID OPTION SPECIFIED

A CPT configuration entry was found with an option field.

Action Required

The configuration macros should not allow this to successfully assemble. Verify that the macros have not been modified and that the assembly was successful.

Source Module

mod_name

CPT607E

mod_name QNAME OPTION NOT SUPPORTED

A CPT configuration entry was found with an invalid option field which is not currently supported.

Action Required

Specify other options until support is available.

Source Module

mod_name

CPT608E

mod_name *ffff* OPTION WRITEQ TD FAILED FOR
QUEUE=*name*

A CPT routine wrote to a transient data queue and it failed. *ffff* is FILE, SEP, ALL, or LL to indicate the type of option being processed.

Action Required

Determine the problem with the transient data queue. Ensure that it is properly configured to CPT and in the CICS DCT.

Source Module

mod_name

200
24030-200100

CPT609E

mod_name *ffff* OPTION PROCESSING API RECV REQUEST
RETURNED ZERO BYTES

A CPT routine received zero bytes from the API. This is a logic error. *ffff* is FILE, SEP, ALL, or LL to indicate the type of option being processed.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

CPT610E

mod_name *ffff* OPTION PROCESSING DETERMINED ZERO
BYTES DURING EOF PROCESSING

The CPT receive transaction determined that a session was opened and closed with zero bytes received. *ffff* is FILE, LL, ALL, or SEP.

Action Required

Correct the sending host to properly send the data.

Source Module

mod_name

CPT611E

***mod_name* LL OPTION PROCESSING RECEIVED A VALUE OF ZERO IN RECLEN FIELD**

The CPT receive transaction determined that a length field contained a value of zero. This is not allowed.

Action Required

Correct the sending host to properly send the data.

Source Module

mod_name

CPT612E

mod_name* LL OPTION PROCESSING RECEIVED A VALUE THAT EXCEEDS TRANSIENT BUFFER; TDSIZE=*nnnnn*, LLSIZE=*lllll

The CPT receive transaction determined that a length field contained a value greater than the transient data buffer. This is not allowed. The TDSIZE is *nnnnn*. The LLSIZE is *lllll*.

Action Required

Correct the sending host to properly send the data or increase the size of the transient data buffer.

Source Module

mod_name

20124030-200100

CPT613E

mod_name SEPARATOR CHAR. NOT FOUND IN LAST RECORD

The CPT routines encountered end of file in *SEP* mode with out finding a separator character. The record is discarded.

Action Required

Correct the sending host to properly send the data.

Source Module

mod_name

CPT614E

mod_name SEPARATOR CHAR. LENGTH FIELD IN CONFIG IS ZERO

The configuration entry specified *SEP*, but the separator length was zero.

Action Required

The configuration macros should not allow this to successfully assemble. Verify that the macros have not been modified and that the assembly was successful.

Source Module

mod_name

200801-024030-200100

CPT615E

mod_name SEPARATOR CHAR. LENGTH FIELD IN CONFIG
EXCEEDS MAX OF 2

The configuration entry specified *SEP*, but the separator length was greater than two.

Action Required

The configuration macros should not allow this to successfully assemble. Verify that the macros have not been modified and that the assembly was successful.

Source Module

mod_name

CPT616E

CPTTREC *ffff* OPTION PROCESSING TRANSIENT BUFFER
OVERRUN (*nnnnn*)

The CPT routines processed a record which was greater than the transient data queue buffer. The record is discarded. *ffff* is *FILE*, *LL*, or *SEP*. *nnnnn* is the buffer length in hex.

Action Required

Configure a larger buffer or correct the sending host to properly send the data.

Source Module

CPTTREC

20
24030-200100

CPT618E

mod_name API TRANSLATION ROUTINE FAILED,
RC=rrrrrrrrr DC=ddddddd

A CPT transaction encountered an error from the CPT translate routine. The return code is *rrrrrrrrr*. The diagnostic code is *ddddddd*.

Action Required

Refer to the CPT translate service return code and diagnostic code information for an explanation.

Source Module

mod_name

CPT620E

mod_name WRITEQ TD BAD BUFFER ADDRESS (ZERO)

A CPT transaction tried to write to a CICS transient data queue and determined the address of the buffer was zero. This is a logic error.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

CPT621E

mod_name WRITEQ TD BAD BUFFER LENGTH (ZERO)

A CPT transaction tried to write to a CICS transient data queue and determined the length of the buffer was zero. This is a logic error.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

CPT622E

mod_name API RECV ROUTINE CALLED WITH BAD ADDRESS (ZERO)

A CPT routine was called with a zero address in the parameter block it was passed. This is a logic error.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

20
024030-200100

CPT623E

mod_name API RECV ROUTINE CALLED WITH BAD LENGTH
(ZERO)

A CPT routine was called with a zero length in the parameter block it was passed. This is a logic error.

Action Required

Correct the length field or contact Interlink Customer Support.

Source Module

mod_name

CPT626E

mod_name QNAME XXXX NOT FOUND IN CONFIG TABLE

The queue name returned by CICS could not be found in the CPTCONFIG module.

Action Required

Correct the CPT configuration or the CICS configuration so the queue names match.

Source Module

mod_name

CPT627E

mod_name CPT SEND ROUTINE REQUEST CONTAINED AN
INVALID BUFFER ADDRESS (ZERO)

The CPT routines were passed a SEND request with a buffer address of zero.

Action Required

Correct the SEND request parameter block or contact Interlink Customer Support.

Source Module

mod_name

CPT628E

mod_name CPT SEND ROUTINE REQUEST CONTAINED AN
INVALID BUFFER LENGTH (ZERO)

The CPT routines were passed a LENGTH request with a buffer address of zero.

Action Required

Correct the SEND request parameter block or contact Interlink Customer Support.

Source Module

mod_name

20-24030-200100

CPT630E

mod_name READQ TD RETURNED BAD BUFFER POINTER
(ZERO)

A CPT transaction issued a CICS READQ for transient data and CICS returned a buffer pointer of zero.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

CPT631E

mod_name READQ TD RETURNED BAD BUFFER LENGTH
(ZERO)

A CPT transaction issued a CICS READQ for transient data and CICS returned a length of zero.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

CPT633E

***mod_name* CPT CONNECT ROUTINE RETURNED AN INVALID
TOKEN**

A CPT transaction issued a `CONNECT` request and it returned an invalid token.
This is a logic error.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

CPT634E

***mod_name ffff* OPTION PROCESSING READQ WAS
SUCCESSFUL; HOWEVER, NO DATA WAS RETURNED**

A CPT transaction issued a CICS `READQ` for transient data, but no data was
returned. *ffff* is `FILE`, `ALL`, `LL`, or `SEP`.

Action Required

Contact Interlink Customer Support.

Source Module

mod_name

200-24030-200100

CPT635E

mod_name ffff OPTION PROCESSING SENT (CLIENT)
TRANSACTION FAILED

An error was determined while processing the SEND tool transaction.

Action Required

Review previous error message to determine cause of error.

Source Module

mod_name

CPT637E

mod_name INITIATED WITH INVALID DATA INPUT (TRANS
QUEUE NAME)

A CPT transaction attempted to determine transient data queue name;
however no input was enter or the input string was invalid.

Action Required

Verify that transaction was initiated with input or that the input string was a valid
transient queue name of 4 characters.

Source Module

mod_name

CPT689E

CPT IVP LOOPBACK TEST FAILED, SEND AND RECV DATA NOT EQUAL

The CPT *Installation Verification Program (IVP)* determined that the text sent was not equal to the text received. This indicates a customization error or a communication error.

Action Required

Review customization of `LISTEN`, `RECEIVE` and `SEND` tool. Verify CPT Error log for additional information. The CPT transaction and API trace options can be used to debug problem.

Source Module

CPTTIPCK

CPT690E

CPT IVP INVALID INPUT ENTERED, DATA =

The CPT IVP was initiated with invalid transaction input.

Action Required

Review syntax of IVP transaction command.

Source Module

CPTTIPCK

CPT691E

CPT IVP RECEIVE FAILED, LENGTH ERROR (MAX LENGTH 80)

The CPT IVP was initiated with invalid transaction input. The input string was greater than 80 characters.

Action Required

Review syntax of IVP transaction command.

Source Module

CPTTIPCK

CPT692E

CPT IVP DELETEQ FOR SEND TD QUEUE FAILED

The CPT IVP `DELETEQ TD` command for the send transient data queue failed.

Action Required

Validate the send transient data queue in the CICS DCT and review syntax of IVP transaction command.

Source Module

CPTTIPCK

CPT693E

CPT IVP DELETEQ FOR RECV TD QUEUE FAILED

The CPT IVP `DELETEQ` TD command for the receive transient data queue failed.

Action Required

Validate the receive transient data queue in the CICS DCT and review syntax of IVP transaction command.

Source Module

CPTTIPCK

CPT694E

CPT IVP RECEIVE FAILED, MISCELLANEOUS ERROR

The CPT IVP `RECEIVE` command for the input failed.

Action Required

Review IVP command syntax or report error to system administrator.

Source Module

CPTTIPCK

CPT695E

CPT IVP READQ TD QUEUE, MISCELLANEOUS ERROR

The CPT IVP `READQ` TD command failed.

Action Required

Review CICS log and verify transient data queue in the CICS DCT.

Source Module

CPTTIPCK

20
24030-200100

CPT696E

CPT IVP READQ TD QUEUE LENGTH ERROR

The CPT IVP READQ TD command failed with a length error.

Action Required

Review CICS log and verify transient data queue in the CICS DCT.

Source Module

CPTTIPCK

CPT697E

CPT IVP WRITEQ TD QUEUE ERROR

The CPT IVP WRITEQ TD command failed.

Action Required

Review CICS log and verify transient data queue in the CICS DCT.

Source Module

CPTTIPCK

CPT698E

CPT IVP TIME OUT, NO DATA RECEIVED WITHIN 30 SECONDS

The CPT IVP failed to complete in 30 seconds. This indicates a customization or communication error.

Action Required

Review customization of `LISTEN`, `RECEIVE` and `SEND` tool. Verify CPT Error log for additional information. The CPT transaction and API trace options can be used to debug problem.

Source Module

CPTTIPCK

CPT699I

CPT IVP LOOPBACK TEST SUCCESSFUL!!!

The CPT IVP found that the data it sent matched the data it received.

Action Required

This is a normal message from the installation verification. program.

Source Module

CPTTIPCK

20
24030-200100

CPT801I

MODULE *mod-name* ENTERED AT HH:MM:SS YY:DDD

A module has been entered in the CPT routines. *mod_name* is the module name
HH:MM:SS is the current time, and *YY:DDD* is the current Julian date. This
message is produced by specifying the `TRNTRAC` or `APITRAC ENTER` keyword.

Action Required

This message indicates the time and date a CPT module was entered.

Source Module

mod_name

CPT802I

LISTENING ON PORT *nnnnn*

The CPT listen transaction is running. *nnnnn* is the port on which it is listening.
This message is produced by specifying the `APISTAT`

Action Required

This message indicates that a server connection is established and listening
for client connections on port *nnnnn*.

Source Module

CPTCLSTN

CPT803I

mod_name CONNECTED TO LOCAL PORT *lllll* FROM
REMOTE PORT *rrrrr*, REMOTE HOST *rhost_name*

A CPT client or server connection has been established on local port *lllll*, remote port *rrrrr* and the remote host name is *rhost_name*. This message is produced by specifying the `APISTAT CONN` keyword.

Action Required

This message indicates information about a newly established client or server connection.

Source Module

mod_name

20
24030-200100

CPT804I

**SEND TRANS (OPTN) STATS - APIBYT=nnnn APISND=nnnn
APIMAX=nnnn RTDBYT=nnnn RTD=nnnn RTDMAX=nnnn**

The client or send transaction statistics are listed by this message. The `OPTN` indicates the configuration option for this queue. This message is produced by specifying the `TRNSTAT TERM` keyword.

PARAMETER	DESCRIPTION
APIBYT	Number of bytes sent to the API interface.
APISND	Number of API send requests issued.
APIMAX	Maximum number of bytes sent to the API interface by any single request.
RTDBYT	Number of bytes read from the CICS transient data queue.
RTD	Number of CICS READQ TD issued.
RTDMAX	Maximum number of bytes read from the transient data queue by any single request.

Action Required

This message indicates resource utilization by the `SEND` tool. This information can be used to tune a given automated service.

Source Module

CPTTSEND

CPT805I

RECV TRANS (OPTN) STATS - APIBYT=nnnn APIRCV=nnnn
APIMAX=nnnn WTDBYT=nnnn WTD=nnnn WTDMAX=nnnn

The server or receive transaction statistics are listed by this message. The OPTN indicates the configuration option for this queue. This message is produced by specifying the TRNSTAT TERM keyword.

PARAMETER	DESCRIPTION
APIBYT	Number of bytes received from the API interface.
APIRCV	Number of API receive requests issued.
APIMAX	Maximum number of bytes received from the API interface by any single request.
WTDBYT	Number of bytes written to the CICS transient data queue.
WTD	Number of CICS WRITEQ TD issued.
WTDMAX	Maximum number of bytes written to the transient data queue by any single request.

Action Required

This message indicates resource utilization by the RECEIVE tool. This information can be used the tune a given automated service.

Source Module

CPTTREC

20124030-200100

CPT806I

MODULE *mod_name* TERMINATED AT *HH:MM:SS YY:DDD*

A module has been exited in the CPT routines. *mod_name* is the module name, *HH:MM:SS* is the current time, and *YY:DDD* is the current Julian date. This message is produced by specifying the `TRNTRAC` or `APITRAC TERM` keyword.

Action Required

This message indicates the time and date a CPT module was terminated.

Source Module

mod_name

CPT807I

**API RECV STATS - APIRCV=*nnnn* RCVBYT=*nnnn*
RTNBYT=*nnnn* RCVTPL=*nnnn* LSTDATA=*nnnn***

CPT TRUE data transfer statistics for a connection. This message is produced by specifying the `APISTAT TERM` keyword.

PARAMETER	DESCRIPTION
APIRCV	Number of API receive routine requests.
RCVBYT	Number of bytes successfully received from the transport provider.
RTNBYT	Number of bytes returned to the calling routine.
RCVTPL	Number of API TRECvs issued.
LSTDATA	Number of bytes unsuccessfully received transport provider.

Action Required

This message indicates resource utilization by the `RECEIVE` service. This information can be used to tune either an automated tool or a user-written application.

Source Module

CPTCCLOS

CPT808I

**API SEND STATS - APISND=nnnn SNDBYT=nnnn
REQBYT=nnnn SNDTPL=nnnn LSTDATA=nnnn**

API send endpoint data statistics for a CICS connection. This message is produced by specifying the `APISTAT TERM` keyword.

PARAMETER	DESCRIPTION
APISND	Number of API send routine requests.
SNDBYT	Number of bytes successfully sent to the transport provider.
REQBYT	Number of bytes requested to be sent by the calling routine.
SNDTPL	Number of API TSENDS issued.
LSTDATA	Number of bytes unsuccessfully sent to the transport provider.

Action Required

This message indicates resource utilization by the `SEND` service. This information can be used to tune either an automated tool or a user-written application.

Source Module

CPTCCLOS

CPT809I

API LISTEN STATS - APIOPN=nnnn TRANS=nnnn BAD
TRANS=nnnn REJECT=nnnn

The server/listen transaction statistics are listed by this message. This message is produced by specifying the `APISTAT TERM` keyword.

PARAMETER	DESCRIPTION
APIOPN	Number of API TOPENS issued for new connections.
TRANS	Number of CICS START TRANSACTIONS issued for new connections.
BAD TRANS	Number of failed CICS START TRANSACTIONS.
REJECT	Number of API TREJECTs issued for failed connections.

Action Required

This message indicates resource utilization by the `LISTEN` (server) service. This information can be used to tune either an automated tool or a user-written application.

Source Module

CPTCCLOS

CPT810I

mod_name INITIAL WRITE TO STATISTICS LOG TD QUEUE

CPT initialization has successfully verified statistics log transient data queue is accessible.

Action Required

This is a normal CPT initialization message.

Source Module

mod_name

CPT901I

mod_name SERVICE LEVEL ENTERED AT TIME
julian_date

This message is generated upon entry to a called CPT module. It is produced by specifying the `APITRAC_ENTR` keyword.

Action Required

Use to verify program flow and/or the PTF level of the CPT installed.

Source Module

mod_name

CPT902I

mod_name HEX DUMP OF CPT CONNECTION MANAGEMENT
ARGUMENT FOLLOWS

This is a hex dump of the *Argument for Connection Management (ACM)*. This message is produced by specifying the `APITRAC` or `TRNTRAC ARG` keyword.

Action Required

This message is generated upon entry into a CPT TRUE and can be used for debugging.

Source Module

mod_name

CPT903I

mod_name GIVE/TAKE ARGUMENT FOLLOWS

This is a hex dump of the *Argument for Facility Management (AFM)*. This message is produced by specifying the `APITRAC ARG` keyword.

Action Required

This message is generated upon entry into a CPT TRUE and can be used for debugging.

Source Module

mod_name

CPT904I

mod_name GIVE/TAKE RETURN ARGUMENT FOLLOWS

This is a hex dump of the AFM. This message is produced by specifying the `APITRAC ARG` keyword.

Action Required

This message is generated upon exit from a CPT TRUE and can be used for debugging.

Source Module

mod_name

CPT905I

LISTEN TERMINATING ON PORT *nnnnn*

The CPT **LISTEN** task is no longer listening on local port number *nnnnn*. This message is produced by specifying the **APITRAC TERM** keyword.

Action Required

This message is generated during listen service termination.

Source Module

mod_name

CPT906I

**CLOSE UNCHAINING. HEX DUMP OF QUEUE HDR AND TOKEN
FOLLOWS**

The CPT **CLOSE** routine is taking a control block off a queue, the queue header is dumped, followed by the control block being unchained, followed by the chain of control blocks. This message is produced by specifying the **APITRAC CLOSE** keyword.

Action Required

This message is generated during connection termination and can be used for debugging.

Source Module

mod_name

20
024030-200100

CPT907I

mod_name HEX DUMP OF CPTMSEND CONFIGURATION ENTRY
FOLLOWS

The CPT `SEND` tool routine is dumping configuration information. This message is produced by specifying the `TRNTRAC ARG` keyword.

Action Required

This message is generated upon entry into the automated `SEND` tool and can be used for debugging.

Source Module

mod_name

CPT908I

mod_name TRANSACTION TERMINATION AT HH:MM:SS

The CPT automated transaction has ended at the specified time. This message is produced by specifying the `TRNTRAC TERM` keyword.

Action Required

This message is generated before termination of an automated transaction and can be used for debugging.

Source Module

mod_name

CPT909I

mod_name HEX DUMP OF CPT TOKEN FOLLOWS

This is a hex dump of the connection token. This message is produced by specifying the `API TRAC TOKEN` keyword.

Action Required

This message is generated upon entry into an automated transaction or CPT TRUE and can be used for debugging.

Source Module

mod_name

CPT910I

mod_name HEX DUMP OF CPTMLSTN CONFIGURATION ENTRY FOLLOWS

The CPT `LISTEN` tool routine is dumping configuration information. This message is produced by specifying the `TRNTRAC ARG` keyword.

Action Required

This message is generated upon entry into the automated `LISTEN` tool and can be used for debugging.

Source Module

mod_name

2024030-200100

CPT911I

mod_name HEX DUMP OF CPT CLOSE ROUTINE ARGUMENT
FOLLOWS

This is a hex dump of the *Argument for Close (ACL)*. This message is produced by specifying the `API TRAC` or `TRN TRAC ARG` keyword.

Action Required

This message is generated upon entry into a CPT TRUE and can be used for debugging

Source Module

mod_name

CPT912I

mod_name HEX DUMP OF CPT CLOSE ROUTINE RETURN
ARGUMENT FOLLOWS

This is a hex dump of the ACL. This message is produced by specifying the `API TRAC ARG` keyword.

Action Required

This message is generated upon exit from a CPT TRUE and can be used for debugging

Source Module

mod_name

CPT913I

mod_name HEX DUMP OF API TREC V TPL DATA FOLLOWS

This is a hex dump of the data received from the transport provider. This message is produced by specifying the TRNTRAC RECV keyword.

Action Required

This message is generated after every transport provider receive command and can be used for debugging.

Source Module

mod_name

CPT914I

mod_name HEX DUMP OF API TSEND TPL DATA FOLLOWS

This is a hex dump of the data sent to the transport provider. This message is produced by specifying the TRNTRAC SEND keyword.

Action Required

This message is generated after every transport provider send command and can be used for debugging.

Source Module

mod_name

20124030-200100

CPT915I

***mod_name* HEX DUMP OF CPT CONNECTION MANAGEMENT
RETURN ARGUMENT FOLLOWS**

This is a hex dump of the ACM. This message is produced by specifying the
API TRAC ARG keyword.

Action Required

This message is generated upon exit from a CPT TRUE and can be used for
debugging.

Source Module

mod_name

CPT917I

***mod_name* HEX DUMP OF API TRANSPORT PARAMETER LIST
FOLLOWS**

This is a hex dump of the API transport provider parameter list (TPL). This
message is produced by specifying the TRNTRAC TPLERR keyword or when an
unknown transport provider error is encountered.

Action Required

This message is generated after an API request error has occurred and is used
for debugging.

Source Module

mod_name

CPT918I

***mod_name* HEX DUMP OF WRITEQ TD DATA FOLLOWS**

This is a hex dump of the data written to the transient data queue. This message is produced by specifying the `TRNTRAC WDT` keyword.

Action Required

This message is generated before every `WRITEQ TD` command and can be used for debugging.

Source Module

mod_name

CPT919I

***mod_name* GIVE UNCHAINING FROM TASK WORK AREA.
QUEUE HDR AND CEPT FOLLOW:**

The CPT `GIVE` routine is taking a *CICS End Point (CEP)* off of a CICS task's work area queue. The queue header (containing the CEP to be unchained) and the current task work area CEP queue is dumped. This message is produced by the `APITRAC=PASS` keyword.

Action Required

This message is generated for the `GIVE` service when the CEP is still on the previous task's work area queue.

20
24030-200100

CPT920I

mod_name TAKE UNCHAINING FROM TASK WORK AREA.
QUEUE HDR AND CEPT FOLLOW:

The CPT TAKE routine is taking a CEP off of a CICS task's work area queue. The queue header (containing the CEP to be unchained) and the current Task Work Area CEP queue is dumped. This message is produced by the APITRAC=PASS keyword.

Action Required

This message is generated for the TAKE service when the CEP is still on the previous task's work area queue.

CPT921I

mod_name HEX DUMP OF CPT DATA TRANSFER ARGUMENT
FOLLOWS

This is a hex dump of the ADT. This message is produced by specifying the APITRAC or TRNTRAC ARG keyword.

Action Required

This message is generated upon entry into a CPT TRUE and can be used for debugging.

Source Module

mod_name

CPT922I

***mod_name* HEX DUMP OF CPT DATA TRANSFER DATA
FOLLOWS**

This is a hex dump of the data transfer data. This message is produced by specifying the `API TRAC` or `TRNTRAC ARG` keyword.

Action Required

This message is generated upon entry into a CPT TRUE and can be used for debugging.

Source Module

mod_name

CPT923I

***mod_name* HEX DUMP OF READQ TD FOLLOWS**

This is a hex dump of the data read from the transient data queue. This message is produced by specifying the `TRNTRAC RDT` keyword.

Action Required

This message is generated after every `READQ TD` command and can be used for debugging.

Source Module

mod_name

20
124030-200100

CPT924I

***mod_name* HEX DUMP OF CPT DATA TRANSFER RETURN
ARGUMENT FOLLOWS**

This is a hex dump of the ADT. This message is produced by specifying the
APITRAC ARG keyword.

Action Required

This message is generated upon exit from a CPT TRUE and can be used for
debugging.

Source Module

mod_name

CPT925I

***mod_name* HEX DUMP OF CPT DATA TRANSFER RETURN
DATA FOLLOWS**

This is a hex dump of the data transfer data. This message is produced by
specifying the APITRAC ARG keyword.

Action Required

This message is generated upon exit from a CPT TRUE and can be used for
debugging.

Source Module

mod_name

CPT926I

***mod_name* HEX DUMP OF CPT DATA TRANSFER VECTOR
LIST FOLLOWS**

This is a hex dump of the ADT vector list form. This message is produced by specifying the `APITRAC ARG` keyword.

Action Required

This message is generated upon entry into a CPT TRUE and can be used for debugging

Source Module

mod_name

CPT927I

***mod_name* HEX DUMP OF TPL ERROR PROCESSOR
PARAMETER LIST FOLLOWS**

This is a dump of the Transport Provider List which CPT uses to communicate with the Transport Control Provider. This message is produced when there is a TCP error, by specifying the `APITRAC TPLERR` keyword. This dump is designed for internal use.

Action Required

None.

Source Module

mod_name

20124030-200100

CPT930I

***mod_name* HEX DUMP OF CPT XLAT ROUTINE ARGUMENT
FOLLOWS**

This is a hex dump of the *Argument for Translation* (AXL). This message is produced by specifying the `API TRAC XLAT` keyword.

Action Required

This message is generated upon entry into a CPT TRUE and can be used for debugging

Source Module

mod_name

CPT931I

***mod_name* HEX DUMP OF CPT XLAT ROUTINE DATA
FOLLOWS**

This is a hex dump of the data translate data. This message is produced by specifying the `TRNTRAC=XLAT ARG` keyword.

Action Required

This message is generated upon entry to a CPT TRUE and can be used for debugging

Source Module

mod_name

CPT932I

***mod_name* HEX DUMP OF CPT XLAT ROUTINE RETURN
ARGUMENT FOLLOWS**

This is a hex dump of the AXL. This message is produced by specifying the
APITRAC XLAT keyword.

Action Required

This message is generated upon exit from a CPT TRUE and can be used for
debugging

Source Module

mod_name

CPT933I

***mod_name* HEX DUMP OF CPT XLAT ROUTINE RETURN DATA
FOLLOWS**

This is a hex dump of the data translate data. This message is produced by
specifying the TRNTRAC XLAT ARG keyword.

Action Required

This message is generated upon exit from a CPT TRUE and can be used for
debugging

Source Module

mod_name

20
124030-200100

CPT934I

**CPT XLAT ROUTINE TRANSLATION TABLE FIRST 128
BYTES FOLLOWS**

This is a hex dump of the translation table. This message is produced by specifying the `APITRAC XLAT` keyword.

Action Required

This message is generated upon entry to a CPT TRUE and can be used for debugging

Source Module

mod_name

CPT936I

***mod_name* INITIAL WRITE TO TRACE LOG TD QUEUE**

CPT initialization has successfully verified `TRACE` log transient data queue is accessible.

Action Required

This is a normal CPT initialization message.

Source Module

mod_name

ABEND CODES

The CPT common modules are written to handle internal abends by issuing a CICS dump request. The request code indicates the source module involved. Then it returns a return code of 20 which indicates to the interface code to issue a CICS abend with a code of CI00. Therefore the abend/dump codes are as follows:

CODE	MODULE	DESCRIPTION
CI00	CPTFXXX	Abend occurred in Service Routine Stub Routine
CI01	CPTCCLOS	Abend occurred in the CLOSE Service Routine
CI02	CPTCCONN	Abend occurred in the CONNECT Service Routine
CI03	CPTCERR	Abend occurred in the ERROR Log Routine
CI06	CPTCLSTN	Abend occurred in the LISTEN Service Routine
CI08	CPTCRECV	Abend occurred in the RECEIVE Service Routine
CI09	CPTCSEND	Abend occurred in the SEND Service Routine
CI10	CPTCSTAT	Abend occurred in the STAT Log Routine
CI11	CPTCTAKE	Abend occurred in the TAKE Service Routine
CI12	CPTCTERM	Abend occurred in the Termination Support Routine
CI13	CPTCTRAC	Abend occurred in the Trace Log Routine
CI14	CPTCXLAT	Abend occurred in the TRANSLATE Service Routine
CI15	CPTCGIVE	Abend occurred in the GIVE Service Routine

Dump codes CI04, CI05, and CI07 are not issued at this time.

A dump will be produced in the CICS dump file. This dump should be formatted and the error reported to the site administrator.

Glossary

This glossary defines the terms and acronyms referred to in Interlink documentation. The first section provides definitions of terms. The second section is a quick reference list of acronyms and what they stand for. All acronyms included in the second section are defined in the first section.

Definitions

Abnormal End of Task (ABEND)

Termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

Access Control Facility 2 (ACF2)

A facility that provides security and access control on the mainframe for a user.

Access Control ID (ACID)

An alphanumeric value up to eight characters in length that TOP SECRET uses to determine a user's access to system resources. Associated with each unique ACID is a set of resource access authorizations and/or administrative authorizations.

Access Method

A mainframe data management routine that moves data between storage and an I/O device in response to requests made by a program.

Access Method Control Block (ACB)

A control block that links an application program to VSAM or VTAM.

Access Method Services (AMS or IDCAMS)

A multifunction service program that defines VSAM data sets and allocates space for them, converts sequential data sets to key-sequenced data sets, modifies data set attributes in the catalog, reorganizes data sets, facilitates data portability between operating systems, creates backup copies of data sets and indexes, helps make inaccessible data sets accessible, and lists the records of data sets and catalogs.

200801-020080-310100

Access Module Control Block (AMDCB)

A control block used by the NFS server to hold all access method dependent data for the duration of the file usage.

Accessor Environment Element (ACEE)

A description of the correct user, including User ID, current connect group, user attributes, and group attributes. An ACEE is constructed during user identification and verification.

Address Resolution Protocol (ARP)

A broadcast-based method for dynamically translating between IP addresses and physical addresses.

Advanced Function Printing (AFP)

A collection of software and hardware products that provides for quality printing from data processing systems. Facilities include electronic forms, text and graphics merging, bar codes, images, signatures, logos, etc.

Advanced Peer-to-Peer Communication (APPC)

An IBM protocol that is part of the SNA architecture that enables structured conversations between two networked applications. Also referred to as a Logical Unit (LU) Type 6.2. (See **Logical Unit**.)

Advanced Peer-to-Peer Networking (APPN)

An extension to the SNA networking architecture that supplies networking functions that are easy to use, are dynamic, and give control of the network to the peer systems that make up the network. APPN lets peer systems participate in and control a network that does not require the control of a traditional SNA host. APPN also handles routing of data by intermediate systems in the network so that all systems become logically adjacent, whether or not they are physically adjacent.

Advanced Resolution Protocol (ARP)

A broadcast-based method for dynamically translating between IP addresses and physical addresses.

American National Standards Institute (ANSI)

An organization that provides a standard specification for languages such as C and COBOL so that systems and programs can be developed to be compatible across platform boundaries.

American Standard Code for Information Interchange (ASCII)

A standard character code used in many computer systems. Traditional ASCII uses only seven bits out of eight possible to represent data, but extended ASCII uses all eight bits to define additional characters.

Application Program Control Block (APCB)

A data structure supplied by the application program as the primary anchor for information required to execute subsequent requests.

Application Program Interface (API)

The formally-defined programming language interface between an IBM system control program or program product and its user.

Architecture

The specific components of a computing system and the way they interact with one another.

ARPANET

A network created by the Defense Advanced Research Projects Agency (DARPA) that connects approximately 150 sites at universities and corporations doing research for the U.S. government. The ARPANET uses the TCP/IP protocol suite. It is part of the Internet.

Attribute Database (ADB)

A catalog database used by the NFS server to hold names of files allocated within the UFS.

Authorized Program Facility (APF)

A security feature of MVS that controls access to authorized system libraries. Programs in these libraries can enter a system-privileged state that permits greater control of internal system processes. These data sets must be pre-defined in order to be used.

Basic Mapping Support (BMS)

A facility that provides CICS commands and options that can be used to format data between an application program and a device in a standard manner.

Berkeley Software Distribution (BSD)

A series of UNIX system implementations developed at the University of California, Berkeley.

Big-endian

A format for storage or transmission of binary data in which the most significant bit (or byte) comes first. The reverse convention is called little-endian.

Binding

A process of associating a variable with an absolute address, identifier, or virtual address, or with a symbolic address or label in a program. (See **Unbinding**.)

Bootting

The process of powering up the computer, testing to determine which attached hardware devices are running, and bringing the operating system kernel into memory and operation.

Build a Directory Entry List (BLDL)

A system service call that obtains a list of information from the directory of a Partitioned Data Set (PDS) or Partitioned Data Set Extended (PDSE).

Bulk Data Transfer (BDT)

A Courier-defined protocol for transferring large amounts of data between a client and server.

CA-TOP SECRET

A security program developed by Computer Associates that validates all user ID and password combinations for sign-on security and all file accesses for data set security.

Caltech Intermediate Form (CIF)

Describes an output file that contains data produced by `cifplot`.

Catalog Entry

A data set containing extensive information required to locate other data sets, to allocate and de-allocate storage space, to verify the access authority of a program or operator, and to accumulate data set usage statistics.

CICS Programmer's Toolkit (CPT)

An application programming interface between CICS application programs and communication subsystems, based on open network protocols.

Client

In a Sun ONC/NFS environment, a machine that uses the resources of the network.

Connection

A communications mechanism or program association that enables interaction between two functions or systems.

Connection Endpoint (CEP)

The termination of one end of a connection within a service access point.

Connection ID (CID)

An eight-bit data item used by VTAM to uniquely identify a session between to Logical Units (LUs).

Connection-Oriented Transport Service (COTS)

A service that provides the means to establish, maintain, and release transport connections providing duplex transmission between two transport users.

Connectionless Network Protocol (CLNP)

The ISO standard for providing connectionless service; the functional equivalent of DARPA IP.

Connectionless Network Service (CLNS)

A service that uses UDP as the transport mechanism.

Connectionless Transport Protocol (CLTP)

Established a TCP connection without going through a specific host.

Connectionless Transport Service (CLTS)

Connectionless-mode service within the transport layer.

Control Vector

In SNA, an indexed data item that accompanies a session control Request Unit (RU). Control vectors can take many different forms, based on the information they contain.

Customer Information Control System (CICS)

An IBM mainframe control system combining elements of database management and data communications, intended to handle transaction-oriented applications. CICS enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases. CICS uses VTAM for communication with devices and programs in the network and VSAM for accessing data sets.

Daemon

A process that handles system-wide functions, such as network administration, or line printer spooling operations. It is a background task that waits for a request for service from a particular port.

Data Control Block (DCB)

A memory area that holds variables used by MVS data management access methods. It is of interest only to Customer Support.

Data Set

For a mainframe environment, a collection of related records stored together and located by means of a catalog entry.

For the ONC/NFS environment, a contiguous string of characters. New lines are recognized, but the record boundaries do not exist.

For the UNIX environment, referred to as a file.

Data Set Organization

The way in which data is arranged within a data set on a mainframe is called the data set organization. Only sequential, direct, partitioned, and VSAM data set organizations are supported by SNS/NFS. (See **Sequential Data Set**, **Key-Sequenced Data Set**, **Direct Data Set**, **Partitioned Data Set**.)

Datagram

The basic unit of information that is passed across the internet. It consists of one or more data packets (e.g., sets of data).

Defense Advanced Research Projects Agency (DARPA)

A military agency whose network, the ARPANET, was the first to use the TCP/IP protocol over its Wide Area Network (WAN). DARPA has sponsored the development of wide area networks and networking software.

Defense Data Network (DDN)

A network that comprises several Department of Defense networks.

Direct-Access Storage Device (DASD)

A device in which access to data is independent of where data resides on the device.

Direct Data Set (DDS)

A data set type used in a mainframe environment for storing data on a random access device that is accessed using a record address. (See **Sequential Data Set** and **Partitioned Data Set**.)

Directory Services Parameter List (DPL)

A list used to pass information for processing directory service requests such as host name resolution, host address resolution, etc.

Disk Operating System (DOS)

An operating system developed by Microsoft for IBM and compatible personal computers.

Document Composition Facility (DCF)

A text processing program. Its main component is the text formatter, called SCRIPT/VS.

Domain

A collection of machines that is administered as a single entity, having its own unique name.

Domain Name Resolver (DNR)

A facility that provides name resolution services to the ACP task group and to other address spaces via cross-memory services.

Domain Name System (DNS)

(See **Network Directory Services**.)

External Gateway Protocol (EGP)

Protocol used by exterior neighbors to advertise reachability information to other autonomous systems.

Endpoint

(See **Transport Endpoint**.)

Enterprise Print Services (EPS)

Software for MVS TCP/IP that provides printer access across the network. EPS lets MVS users send data to printers attached to other systems on the TCP/IP network and other users on other systems on the TCP/IP network send data to printers controlled by the MVS JES2/JES subsystems.

Entity

Open Systems Interconnection (OSI) terminology for a layer protocol machine. An entity within a layer performs the functions of the layer within a single computer system, accessing the layer entity below and providing services to the layer entity above at local service access points.

Entry-Sequenced Data Set (ESDS)

A type of data set used in a mainframe environment. The format consists of logical records sequenced by the time of their arrival. A particular record is located using the Relative Byte Address (RBA).

Ethernet

A commonly-used, local area network technology originally developed by the Xerox Corporation. This is a form of network that provides real-time communication between machines.

Event Control Block (ECB)

An MVS control block used to communicate between MVS services and application or system modules.

Exit

A mechanism that provides a controlled interface from a server application to a user-provided function. Exits are used in SNS/NFS to interface to site security, archival storage, and accounting software.

Expedited Data

(See **Out-of-Band**.)

Extended Binary-Coded Decimal Interchange Code (EBCDIC)

A set of 256 characters consisting of eight-bit coded characters.

External Control Block (XCB)

A data area that contains the variables used by the NFS server task for the interface to MVS data management for a particular file access.

External Data Representation (XDR)

An encoding convention for data transfer between computers on a network. XDR specifies byte order, record padding, etc. in an architecture-independent manner.

Fastpath

An Intel 977x server controller.

FDDI Channel Attachment (FCA)

An implementation for FDDI in 3762.

Fiber Distributed Data Interface (FDDI)

An emerging high-speed networking standard, the underlying medium of which is fiber optics, and the topology is a dual-attached, counter-rotating Token Ring.

File

An ordered collection of data in a UNIX environment stored or processed as a unit. This collection of data is called a data set in a mainframe environment.

File Handle Database (FHDB)

A data set maintained by the NFS server that holds information about workstation mount points across separate executions of the server task.

File Transfer, Access, and Management (FTAM)

The Open Systems Interconnection (OSI) remote file service and protocol.

File Transfer Protocol (FTP)

A protocol (and program) used to transfer files between host machines in a TCP/IP network.

File System

A hierarchical arrangement of directories and files.

File Usage Block (FUB)

An Internal Control Block (ICB) used to contain variables during an individual access (read, write, etc.) on a file within NFS.

GateD

Program that implements dynamic routing protocols such as RIP, HELLO, EGP, and OSPF.

Gateway

A functional unit that interconnects a local data network with another network having different protocols. It enables networks using different protocols to communicate with each other.

Global Storage Block (GSB)

A data area created and used by the NFS user exits. The GSB address is maintained by the NFS server for the duration of its execution.

Global System Options (GSO)

A definition file that defines system options, such as operating and performance parameters, and site exits.

Graphical User Interface (GUI)

A high-level interface that uses windows and menus with graphic symbols instead of typed system commands to provide an interactive environment for a user.

Gratuitous ARP Reply (GARP)

An ARP reply that is issued whenever a driver terminates. This is part Fault Tolerance.

High-level Index

An entry in the system catalog.

IDC Access Method Services (IDCAMS)

(See **Access Method Services**.)

Indexed Data Set

(See **Key-Sequenced Data Set**.)

Information Management System (IMS)

An IBM-licensed product that provides transaction-based processing on MVS.

Infrastructure (IFS)

A software product that provides a multiple thread execution environment for MVS tasks. SNS/TCPaccess uses IFS.

Initial Program Load (IPL)

The process of readying an IBM mainframe system for operation. The IPL loads a series of software programs from disk and initiates system tasks.

Institute of Electrical and Electronic Engineers (IEEE)

An organization that sets standards for computers and communications.

Instruction Length Code (ILC)

On an IBM/370 system, a two-byte memory location that contains the length of the instruction being executed.

Interior Gateway Protocol (IGP)

Protocol used by interior gateways to exchange reachability and routing information.

Interactive Systems Productivity Facility (ISPF)

A TSO utility that lets users edit data sets in full-screen mode, submit jobs, and retrieve output from a 3270 display terminal (or equivalent).

Interface

A shared boundary between two or more entities. This might be a hardware or software component that links two devices or programs together.

Internal Control Block (ICB)

A control block data structure used exclusively by SNS/NFS during its operation. The layout and contents may change without notice and are of interest only to Customer Support.

International Organization for Standards (ISO)

Best known for the seven-layer OSI Reference Model, a standard model of architecture that constitutes the framework for the development of standard protocols. Previously known as the International Standards Organization. (See **Open Systems Interconnection**.)

International Standards Organization (ISO)

(See **International Organization for Standards**)

International Telegraph and Telephone Consultative Committee (CCITT)

An international advisory body whose charter is to develop international standards for telegraph and telephone companies.

Inter-network Protocol (IP)

The TCP/IP layer between the higher level host-to-host protocol and the local network protocols. IP uses local area network protocols to carry packets, in the form of datagrams, to the next gateway, router, or destination host.

Internet

A global web of interconnected university, business, military, and science computer networks. This network of networks consists of tens-of-thousands of computer systems connected to thousands of interconnected networks.

Internet Control Message Protocol (ICMP)

A standard protocol that is an integral part of IP but uses IP as a higher-level protocol. It lets a gateway or destination host communicate with the source host to report errors on any IP datagram with the exception of ICMP messages.

I/O Configuration Program

A program that defines to a system all available I/O devices and channel paths.

Job Control Language (JCL)

A series of instructions to the MVS operating system describing which programs to run and which data sets to associate with those programs.

Job Entry Subsystem (JES)

A system facility for spooling, job queueing, and managing I/O.

Kernel

A master program set of UNIX software that manages all the physical resources of the computer, including file system management, virtual memory, reading and writing files to disks and tapes, scheduling of processes, and printing and communicating over a network.

Key-Sequenced Data Set (KSDS)

A type of data set used in a mainframe environment for storing data on a random access device. The format consists of an index followed by one or more logical records. This is also known as indexed data set.

Linear Data Set (LDS)

A named linear string of data stored in such a way that it can be retrieved or updated in 4096-byte units. A linear data set object is essentially a VSAM entry-sequenced data set that is processed as an unstructured byte stream.

Link State Advertisement (LSA)

A family of OSPF packets that contain information on routes and routers.

Little-endian

A format for storage or transmission of binary data in which the least significant byte (or bit) comes first. (See **Big-endian**.)

Local Area Network (LAN)

A data network located on the user's premises in which serial transmission is used for direct data communication among data stations.

Logical Unit (LU)

In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points. An LU can support at least two sessions and may be capable of supporting many sessions with other LUs. The type of services provided by the LU to the end user differs for each type of LU. LU types 2 and 3 support communication between processes and terminals. LU 6.2 supports general communication between programs in a distributed processing environment. (See **Advanced Peer-to-Peer Communication**.)

Logon ID (LID)

A record that contains the logon identification of the user. This is an ACF2 term.

Mainframe

A computer running the MVS operating system, including, but not limited to, machines from IBM, Amdahl, Fujitsu, and NAS.

Member

A term for an independent group of sequentially-organized records in a partitioned data set used on a mainframe. Message Handling System (MHS)

The system of message user agents, message transfer agents, message stores, and access units that together provide Open Systems Interconnection (OSI) electronic mail.

Mount

A process to access a directory from a local disk attached to the machine making the mount request (UNIX mount) or remote disk on a network (NFS mount).

An operation that associates a group of files on a server with a directory (mount point) on a client to provide transparent access to the files through that directory. The files must be in a hierarchical arrangement.

Mount Point

A place established in a workstation or server local directory that is used during the transparent accessing of a remote file.

Multiple Virtual Storage (MVS)

An IBM-licensed multipurpose operating system for the System/370 and System/390 processors. MVS handles three major functional areas: job management (e.g., command processing and job processing), supervisor management (e.g., resource management), and data management (e.g., file and input/output operations). MVS supports such characteristics as a large number of concurrent users, teleprocessing, and efficient storage and retrieval from large databases.

Network Data Set Header (NDH)

An NJE control record that generally precedes a unit of `SYSDOUT` data. It may also appear in the middle of `SYSDIN` data to indicate a change in the format of the `SYSDIN` data.

Network Directory Services (NDS)

A distributed database system that provides dynamic information retrieval for host-specific address and name information.

Network File System (NFS)

A distributed component of `ONC` developed by Sun Microsystems that lets a set of computers cooperatively access each other's files transparently. Once accessed the filesystem appears to reside on the local host.

Network Information Center (NIC)

A central administration facility for the Internet. The NIC supervises network names and network addresses, and provides several information services.

Network Job Entry (NJE)

A protocol that enables jobs to be submitted and output returned remotely.

Network Service Access Point (NSAP)

The point at which the OSI network service is made available to a transport entity.

Network Virtual Terminal (NVT)

A set of rules defining a very simple virtual terminal interaction. The NVT is used at the start of a TELNET session, but a more complex type of terminal interaction can be negotiated.

Node Initialization Block (NIB)

In VTAM, a control block associated with a particular node or session that contains information used by the application program to identify the node or session and to indicate how communication requests on a session are to be handled by VTAM.

Non-Broadcast Multi-Access (NBMA)

A non-point-to-point interface that is not broadcast-capable.

Open Network Computing (ONC)

A set of software that supports heterogeneous distributed computing.

Open Shortest Path First (OSPF)

Routing protocol for interior gateways that allows dynamic routing choices to be implemented.

Open System

A system that is internally independent but provides common external services via standardized protocols publicly defined for each layer. (See **Open Systems Interconnection**.) Open Systems Interconnection (OSI)

An international standardization program to facilitate communications among computers from different manufacturers. (See **International Standards Organization**.)

Operating System/2 (OS/2)

A multi-tasking operating system for IBM and compatible personal computers, developed by IBM and Microsoft.

Out-of-Band (OOB)

A mechanism in TCP used to bypass flow control and reach the server immediately. Also known as urgent data or expedited data.

Partitioned Data Set (PDS)

A direct access storage data set that is divided into partitions, called members, each of which can contain a program, part of a program, or data. The PDS has a directory that points to the locations of the various members stored in this data set. This directory allows the members to be independently accessed. Partitioned data sets are often used to store libraries of programs and macro instructions. (See **Sequential Data Set**, **Direct Data Set**).

Partitioned Data Set Extended (PDSE)

A storage management subsystem-managed, page-formatted, data set that is divided into partitions called members, each of which can contain non-load module data.

Pipeline

A connection between the standard output of one process and the standard input of a second process.

Portable Operating System Environment for Computer Environments (POSIX)

A family of operating system standards developed by IEEE that defines the services that must be provided to be "POSIX compliant".

Portmapper

A program that maps client programs to the port numbers of server programs. The portmapper is used with Remote Procedure Call (RPC) programs.

Primary Logical Unit (PLU)

In SNA, the Logical Unit (LU) that contains the primary half-session for a particular LU-LU session. Each session must have a PLU and Secondary Logical Unit (SLU). The PLU is the unit responsible for the bind and is the controlling LU for the session. A particular LU may contain both primary and secondary half-sessions for different active LU-LU sessions. (See **Secondary Logical Unit (SLU)**).

Primitive

A pair of functions that provide an unreliable datagram packet transport between two systems.

Process Control Entry (PCE)

The control block that represents an EPS internal process (i.e., an independent thread of execution).

Procedure Correlation Identifier (PCID)

In SNA, a value used by a control point to correlate requests and replies.

Process

A program in operation. For example, a daemon is a system process that is always running on the system. A process is usually logical, not physical.

Program Request Block (PRB)

An internal MVS control block that identifies a user program to the system. This control block is used by MVS task management to dispatch the executable program.

Program Status Word (PSW)

In the 370 architecture, an eight-byte memory location that includes the instruction counter and control flags of a running program. A PSW is part of the state of a running process.

Protocol

A set of rules for communicating between diverse systems. The rules are mutually understood and followed by the different systems or processes. The protocol specifies actions that can be taken by a station when it receives a transmission or detects an error condition.

Protocol Data Unit (PDU)

A generic term for the protocol unit (e.g., a header and data) used at any layer. Referred to as Transport Protocol Data Unit (TPDU) when used within the transport layer.

Prototype

A model suitable for use as a model for developing software or systems.

Provider

(See **Transport Provider**.)

Relative Byte Address (RBA)

The displacement of a data item from the beginning of the data set, independent of the manner in which the data set is stored.

Relative Record Data Set (RRDS)

A type of data set used in the mainframe environment. It must be on a direct access volume and the format consists of one logical record in a fixed-length slot. Each slot has a unique relative record number. Data is placed in a specific slot, based on a user-supplied relative record number.

Remote Procedure Call (RPC)

A facility that a client uses to request the execution of a procedure call from a server. This facility includes a library of procedures and an external data representation.

Request Parameter List (RPL)

A data structure used to request a function of a VSAM data set or VTAM Logical Unit (LU). The contents of an RPL include addresses of data, status codes, and internal control parameters.

Request/Response Unit (RU)

In SNA, a message unit that requests a function or responds to a function. The RU contains flags describing the remaining portion of the message.

Resource Access Control Facility (RACF)

An IBM-licensed program that provides for access control by identifying and by verifying the users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected resources.

Resource Definition Online (RDO)

A facility that allows for dynamic definition of CICS resources.

Root

A privileged user name, typically in a UNIX environment, that gives the user the ability to modify restricted system files. (See **Superuser**).

ROSCOE

A user interface for MVS providing an alternative to TSO.

Routing Information Protocol (RIP)

Vector-distance routing protocol.

Secondary Logical Unit (SLU)

In SNA, the Logical Unit (LU) that contains the secondary half-session for a particular LU-LU session. An LU may contain secondary and primary half-sessions for different active LU-LU sessions. (See **Primary Logical Unit (PLU)**.)

Sequential Data Set

A type of data set used in the mainframe environment. It must be on a direct access volume with its records stored and retrieved according to their physical order within the data set. (See **Direct Data Set**, **Partitioned Data Set**.)

Server

A machine providing resources to the network. It provides a network service, such as disk storage and file transfer, or contains a program that provides such a service.

Service Access Point (SAP)

The point at which the services of an OSI layer are made available to the next higher layer.

Service Data Unit (SDU)

The unit of data transferred across the interface between a user and provides, and subsequently between two peer users.

Set Buffer Address (SBA)

A control sequence in a 3270 data stream that specifies a new display address to associate with the data that follows.

Sharing

A term used in a computing environment to refer to using a file on a remote system. This is done by mounting the remote file system, then reading from or writing to files in that remote system.

Sim3278/TCPIP Host Program Interface (SIMPCS)

Software manufactured by Simware that is used to enable VT-class terminals to invoke User telnet services in SNS/TCPACCESS.

Simple Mail Transfer Protocol (SMTP)

A protocol that defines how mail is sent in a TCP/IP environment.

SNA Character String (SCS)

In SNA, a character string composed of EBCDIC controls, optionally intermixed with end-user data, that is carried within a request/response unit. Controls EBCDIC control codes in a printer data stream to format printed pages, set modes of device operation, define unique data, or communicate between a user and an application program.

SNS/TCPaccess

A communication subsystem for the internet protocols. It runs on an IBM 370-style mainframe under the MVS operating system.

Socket

An end point for communication that can be named and addressed in a network. The socket interface is one of several Application Program Interfaces (APIs) to the communication protocols.

Software Network Solution (SNS)

The Interlink family of software products that enables communication between networks and mainframes.

Spool Display and Search Facility (SDSF)

An IBM-licensed product that permits time-sharing users to browse and administer the JES2 spool data sets.

Started Task

An MVS system-related job that is initiated by operators from their consoles.

Started Task Control (STC)

A task started from the operator console of a system.

Storage Management System (SMS)

An IBM program product that provides hierarchic management of mainframe disk storage resources.

Subnet

A networking scheme that divides a single logical network into smaller physical networks to simplify routing.

Superuser

A user with special privileges granted if the correct password is supplied when logging in as root or using the *su* command.(See **Root**.)

Supervisor Call (SVC)

The instruction used by the mainframe to generate a software interrupt. Control is then transferred to a routine that handles the interrupt processing. SVC 99 is used to dynamically allocate and deallocate data sets.

System Authorization Facility (SAF)

A system that gets control in response to a request from a resource manager. SAF conditionally directs control to RACF or an installation-supplied processing routine if present, or both.

System Catalog

The highest level catalog on a mainframe and must exist so that the operating system can find data files. It is a VSAM data set and can contain pointers to VSAM data sets, VSAM user catalogs, OS data sets, and OS user catalogs.

System Diagnostic Work Area (SDWA)

An MVS control block that is passed to an abnormal termination recovery routine by the operating system. The SDWA contains the system state at the time of the failure.

System Management Facility (SMF)

A function used on the mainframe to log accounting information, including processor time, data transfer statistics, and user information.

System Modification Program (SMP)

An operating system component that facilitates the process of installing and servicing an MVS system. (See **System Modification Program/Extended**.)

System Modification Program/Extended (SMP/E)

An IBM system programming utility that manages and installs software components into the MVS operating system environment. (See **System Modification Program**.)

System Network Architecture (SNA)

An IBM-licensed network architecture that defines the rules for synchronous communication between two devices in a network. SNA describes the logical structure, formats, protocols, and operational sequences for transmitting information units through networks, and for controlling the configuration and operation of networks.

System Services Control Point (SSCP)

In SNA, a central location point within an SNA network for managing the configuration, coordinating network operation and problem determination requests, and providing directory support and other session services for end users of the network.

TELNET

The virtual terminal protocol in the Internet suite of protocols. Lets a user of one host log into a remote host and interact as a normal terminal user of that host.

Time Sharing Option (TSO)

An MVS operating system feature that provides interactive execution of programs and time sharing from remote terminals.

Token

In a local area network, the symbol of authority that is passed from one data station to another. The token indicates the data station that currently has transmission control on the network.

TOP SECRET

(See **CA-TOP SECRET**.)

Transmission Control Protocol (TCP)

The TCP/IP layer that provides reliable, process-to-process data stream delivery between nodes in interconnected computer networks. TCP assumes that IP is the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP)

A suite of protocols designed to allow communication between networks regardless of the technologies implemented in each network. This connectivity includes the ability to transfer files, send mail, and log on to a remote host in a network of heterogeneous systems. TCP/IP forms the base technology for a large internet that connects most major research institutions, including university, corporate, and government labs.

Transport

A method by which data is moved across a network. The Transport Layer Interface (TLI) lets programming interfaces be built, which ensure that the transport mechanism transmits data to the network reliably.

Transport Connection

A connection between two transport users.

Transport Connection Endpoint (TCEP)

The termination of a connection between two transport users within a TSAP.

Transport Endpoint

(See **Transport Connection Endpoint (TCEP)**.)

Transport Layer

The Open Systems Interconnection (OSI) layer that is responsible for reliable end-to-end data transfer between TCP systems. (See **Transmission Control Protocol**.)

Transport Layer Interface (TLI)

A programmable interface that lets applications be built as independent of the networking protocols below them, and provides reliable network transmission.

Transport Protocol Class 0 - 4 (TP0-4)

The five Open Systems Interconnection (OSI) transport protocols. TP0 (Simple Class) is the simplest and is useful only on top of an X.25 network (or other network that does not lose or damage data). TP4 (Error Detection and Recovery Class) is the most powerful layer and is useful on top of any type of network.

Transport Protocol Data Unit (TPDU)

(See **Protocol Data Unit (PDU)**.)

Transport Provider

The interface that allows application programs access to network transport services.

Transport Provider Address Space (TPAS)

The address space containing the SNS/TCPaccess API subsystem and the transport service provider that processes a service request.

Transport Service Access Point (TSAP)

A communication path between a transport user and a specific transport provider.

Transport Service Data Unit (TSDU)

See **Service Data Unit (TSDU)**.)

Transport Service Parameter List (TPL)

The primary structure for requesting API services and exchanging information between the application program and the transport services.

Transport Services Information Block (TIB)

A data area that specifies parameter values of the API TCP and UDP transport services.

Transport User Address Space (TUAS)

The address space of an application program initiating a service request.

Unbinding

A function that removes the association of a variable with the address or point to which it is bound. (See **Binding**.)

UNIX Filesystem (UFS)

An NFS feature that enables storage of UNIX-style files (byte streams) in MVS storage volumes. UFS files are not translated or reformatted.

Urgent Data

(See **Out-of-Band**.)

User

A person or program function that uses the services provided by a server. A host can be a user and a server at the same time.

User Datagram Protocol (UDP)

A datagram-level protocol built directly on the IP layer. UDP is used for application-to-application programs between TCP/IP hosts. NFS uses UDP as its low-level transport protocol.

User ID (UID)

A token that identifies a user to an operating system. In the UNIX environment, the UID is a numeric value. MVS generally refers to the user identification value as a username.

User Storage Block (USB)

A data area created and used by the NFS user exits. The USB address is maintained by the NFS server for the duration of a user session.

Virtual Machine (VM)

An IBM-licensed operating system that manages the resources of a single computer so that multiple computing systems appear to exist. Each virtual machine is the functional equivalent of a physical (i.e., "real") machine.

Virtual Memory System (VMS)

An interactive operating system that lets you conduct a dialog of command and response with the system.

Virtual Printer (VPRT)

A software simulation of a physical printer that responds to the same print formats as an actual printer. In EPS, the virtual printer subtask is called VPRT.

Virtual Storage Access Method (VSAM)

An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

Virtual Telecommunications Access Method (VTAM)

An IBM-licensed program that controls communication between nodes and application programs, as well as the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability, and lets application programs communicate with other programs and devices in a network.

Virtual Terminal (VT)

An intelligent device, such as a personal computer, that appears to be a terminal to the host computer to which it is connected.

Volume Table of Contents (VTOC)

A table on a direct access volume that describes each data set on the volume.

Wide Area Network (WAN)

A network that consists of machines connected over a wide geographic area. The ARPANET is an example of a wide area network.

Workstation

Any machine or computer on a network that implements communication between enhanced connectivity programs. These communicating programs reside on both the workstation and host system.

Write to Operator with Reply (WTOR)

An MVS system services function that presents a message to the system operator and allows a reply.

Acronyms

ABEND

Abnormal End of Task

ACB

Access Method Control Block

ACF2

Access Control Facility 2

ACEE

Accessor Environment Element

ACID

Access Control ID

ADB

Attribute Database

AFP

Advanced Function Printing

AMDCB

Access Module Control Block

AMS

Access Method Services

ANSI

American National Standards Institute

APCB

Application Program Control Block

APF

Authorized Program Facility

API

Application Program Interface

APPC

Advanced Peer-to-Peer Communication

APPN

Advanced Peer-to-Peer Networking

ARP

Address Resolution Protocol

ASCII

American Standard Code for Information Interchange (from ANSI)

BDT

Bulk Data Transfer

BLDL

Build a Directory Entry List

BMS

Basic Mapping Support

BSD

Berkeley Software Distribution

CCITT

International Telegraph and Telephone Consultative Committee

CEP

Connection Endpoint

CICS

Customer Information Control System

CID

Connection ID

CIF

Caltech Intermediate Form

CLNP

Connectionless Network Protocol

CLNS

Connectionless Network Service

CLTP

Connectionless Transport Protocol

CLTS

Connectionless Transport Service

COTS

Connection-Oriented Transport Service

CPT

CICS Programmer's Toolkit

DARPA

Defense Advanced Research Projects Agency

DASD

Direct-Access Storage Device

DCB

Data Control Block

DCF

Document Composition Facility

DDN

Defense Data Network

DDS

Direct Data Set

DNR

Domain Name Resolver

DNS

Domain Name System

DOS

Disk Operating System

DPL

Directory Services Parameter List

EBCDIC

Extended Binary Coded Decimal Interchange Characters (from IBM)

ECB

Event Control Block

EGP

External Gateway Protocol

EPS

Enterprise Print Services

ESDS

Entry-Sequenced Data Set

FCA

FDDI Channel Attachment

FDDI

Fiber Distributed Data Interface

FHDB

File Handle Database

FTAM

File Transfer, Access, and Management

FTP

File Transfer Protocol

FUB

File Usage Block

GARP

Gratuitous ARP Reply

GSB

Global Storage Block

GSO

Global System Options

GUI

Graphical User Interface

ICB

Internal Control Block

ICMP

Internet Control Message Protocol

IDCAMS

IDC Access Method Services

IEEE

Institute of Electrical and Electronic Engineers

IFS

Infrastructure

ILC

Instruction Length Code

IMS

Information Management System

IGP

Internal Gateway Protocol

IOPC

I/O configuration Program

IP

Inter-network Protocol

IPL

Initial Program Load

ISOInternational Organization for Standards
(previously known as International Standards
Organization)**ISPF**

Interactive System Productivity Facility

JCL

Job Control Language

JES

Job Entry Subsystem

KSDS

Key-Sequenced Data Set

LAN

Local Area Network

LDS

Linear Data Set

LID

Logon ID

LSA

Link State Advertisement

LU

Logical Unit

MHS

Message Handling System

MVS

Multiple Virtual Storage

NBMA

Non-Broadcast Multi-Access

NDH

Network Data Set Header

NDS

Network Directory Services

NFS

Network File System

NIB

Node Initialization Block

NIC

Network Information Center

NJE

Network Job Entry

NVT

Network Virtual Terminal

ONC

Open Network Computing

OOB

Out-of-Band

OS/2

Operating System/2

OSI

Open Systems Interconnection

OSPF

Open Shortest Path First

PCE

Process Control Entry

PCID

Procedure Correlation Identifier

PDS

Partitioned Data Set

PDSE

Partitioned Data Set Extended

PDU

Protocol Data Unit

PLU

Primary Logical Unit

POSIXPortable Operating System Environment for
Computer Environments**PRB**

Program Request Block

PSW

Program Status Word

RACF

Resource Access Control Facility

RBA

Relative Byte Address

RDO

Resource Definition Online

RIP

Routing Information Protocol

RPC

Remote Procedure Call

RPL

Request Parameter List

RRDS

Relative Record Data Set

RU

Request/Response Unit

SAF

System Authorization Facility

SAP

Service Access Point

SBA

Set Buffer Address

SCS

SNA Character Set

SDSF

Spool Display and Search Facility

SDU

Service Data Unit

SDWA

System Diagnostic Work Area

SLU

Secondary Logical Unit

SMF

System Management Facility

SMP

System Modification Program

SMP/E

System Modification Program/Extended

SMS

Storage Management System

SMTP

Simple Mail Transfer Protocol

SNA

System Network Architecture

SNS

Software Network Solution

SPF

Shortest Path First

SSCP

System Services Control Point

STC

Started Task Control

SVC

Supervisor Call

TCEP

Transport Connection Endpoint

TCP

Transmission Control Protocol

TCP/IP

Transmission Control Protocol/Internet Protocol

TIB

Transport Service Information Block

TLI

Transport Layer Interface

TP0 - TP4

Transport Protocol Class 0 through Transport Protocol Class 4

TPL

Transport Service Parameter List

TPAS

Transport Provider Address Space

TPDU

Transport Protocol Data Unit

TSAP

Transport Service Access Point

TSDU

Transport Service Data Unit

TSO

Time Sharing Option

TUAS

Transport User Address Space

20080-310100

UDP

User Datagram Protocol

UFS

UNIX File System

UID

User ID

USB

User Storage Block

VM

Virtual Machine

VMS

Virtual Memory System

VPRT

Virtual Printer

VSAM

Virtual Storage Access Method

VT

Virtual Terminal

VTAM

Virtual Telecommunications Access Method

VTOC

Volume Table of Contents

WAN

Wide Area Network

WTOR

Write To Operator with Reply

XCB

External Control Block

XDR

External Data Representation



200801-020080-310100

C

C

C

C

C

Index

A

abend codes C-126

active task (AMXT) monitoring 3-28

Add/Alter Listen Tool panel

- Administrator Interface usage 4-40 to 4-43
- creating new Listen/Receive Tool definitions 4-40
- field descriptions 4-41 to 4-43
- modifying current parameters 4-40

Add/Alter Send Tool panel

- Administrator Interface usage 4-49 to 4-52
- creating new Send tool definitions 4-49
- field descriptions 4-50 to 4-52
- modifying current parameters 4-49

Administrator Interface

- accessing through the main menu 4-4
- Add/Alter Listen Tool panel 4-40 to 4-43
- Add/Alter Send Tool panel 4-49 to 4-52
- Browse Detail panel 4-12 to 4-19
- bypassing the Main menu 4-5
- configuration 4-30 to 4-53
- Configuration Table Parameters panel 4-30 to 4-34
- CPT API Trace Options panel 4-44, 4-53
- CPT Interface Error Log panel 4-56 to 4-60
- defined 4-3
- definitions of PF keys and navigation codes 4-5
- features 4-3

Generate Trace Entry panel 4-26 to 4-27
leaving 4-5

Listen/Receive Tool panel 4-35 to 4-39

Online Trace Options panel 4-21 to 4-25

Operation Control menu 4-61

Operations component 4-54 to 4-61

Operations Control Menu panel 4-54 to 4-55

Quiesce/Close panel 4-57

Quiesce/Confirm panel 4-18

Reset Trace Online Queue panel 4-28 to 4-29

Send Tool panel 4-45 to 4-47

utilization 4-7, 4-7 to 4-29

Utilization Summary panel 4-7 to 4-11

API trace options

- Listen/Receive tool options, setting 4-44
- setting/removing

 - with the CPT API Trace Options panel 4-53

 - with the Listen/Receive tool 4-44

audience (document) 2

automated CPT transactions processing options 2-3

automatic CPT termination 5-7

B

Browse Detail panel

- Administrator Interface usage 4-12 to 4-19

- confirming the quiesce 4-18

- displaying

 - active connection endpoints 4-12

- client UDP transaction detail 4-17
- cumulative statistics 4-14
- endpoint statistics 4-12, 4-14
- listener endpoints 4-16
- server transaction details 4-13
- UDP endpoints 4-17

- sending an ICMP echo request 4-18
- setting/removing trace options 4-20
- terminating the endpoint task 4-18
- viewing statistical counts 5-9

C

CICS

- components 1-3
- CPT code tools 2-3
- DCT definition statements for transient data queue names 3-14
- executing to verify CPT installation 3-29
- execution JCL, modifying 3-12
- PCT program control statements 3-21, 3-25
- PLT entries for initialization 3-14
- PLTSD table, terminating CPT 1-10
- PPT
 - processing program statements 3-16
 - program statements 3-18
- Programmer's Toolkit overview 1-3
- resource definition tables, adding CPT definition statements 3-13
- restricting CPT startup and shutdown transaction security 5-8
- TD UNIX programs
 - installing with FTP 3-36
 - test programs 3-31

Client

- UDP transaction detail, displaying with the Browse Detail panel 4-17
- Utilization Summary details 4-11

CNTL file, loading from the installation tape 3-4

command notation used in the document 3

configuration

- Administrator Interface 4-30 to 4-53
- customizing CPT configuration file overview 1-9
- modifying the Listen/Receive tool parameters 4-40
- starting the Listen/Receive tool 4-39

Configuration Table

- defining statistics parameters in 4-9
- field descriptions 4-32 to 4-34
- Parameters panel

Administrator Interface usage 4-30 to 4-34

setting cumulative statistics parameters 4-14

CPT

- abend codes C-126
- allocating data sets 3-5
- allocating data sets example 3-5 to 3-7
- API 1-8
 - application interface 1-4
 - communications configuration 1-8
 - overview 1-3
 - subroutine services provided 1-8

collecting transaction statistics 4-14

components

- CPT API subroutines overview 1-3
- tools overview 1-3

configuration file

- Loopback example 3-9 to 3-11
- macro instructions 3-9

cumulative statistics records 5-9

data sets, allocating 3-5

definition statements A-4 to A-18

- adding to CICS definition resource tables 3-13

T09MCICS A-4

T09MEND A-18

T09MLSTN A-6

T09MRECV A-11

T09MSEND A-14

displaying transaction summary statistics 4-9

environment

- CICS resource definitions 1-9

- CPT configuration file 1-9

- customizing using configuration file overview 1-9

- initializing 1-10

- installation and maintenance overview 1-9

- installing and customizing 1-9

- termination program 1-10

Error Log panel, browsing the Error Log queue 4-57

error messages, displaying with the CPT Interface Error Log panel 4-56

halting with the IPPR termination transaction 3-30

initialization program, defining in PLTPI table 3-13

Installation Verification Procedure (IVP) 3-29

Interface

- CICS transaction security 5-8

- confirming closed with the Operations Control Menu panel 4-58

200
24030-200100

- diagnosing and managing 4-54
- failure during initialization 5-5
- initialization messages 5-4
- initializing with the T09TSTRT program 5-4
- starting with the Operations Control Menu 4-61
- terminating 5-6
- terminating automatically 5-7
- termination messages 5-7
- Interface Error Log panel
 - Administrator Interface usage 4-56 to 4-60
 - displaying CPT error messages 4-56
- JCL installation example 3-8
- listing endpoints in the Utilization Summary 4-9
- macros
 - customizing 3-9
 - instructions A-3
- manual pages
 - CL1 (1) library call B-4 to B-5
 - library calls B-1 to B-7
 - SV1 (1) library call B-6 to B-7
- operating environment, maintaining 1-3
- PCT entries from CPT SAMP library 3-21 to 3-23, 3-25 to 3-27
- program names 3-13
- sample program definitions 3-20 to 3-21, 3-27 to 3-28
- termination program, defining in PLTSD table 1-10, 3-13
- testing in loopback mode 3-30
- tools 1-5
 - automated transactions 2-3
 - automatic connection and initialization 1-5
 - CICS code 2-3
 - customizing overview 2-3
 - LISTEN 2-4
 - network interface facility 1-4
 - overview 1-3
 - processing options 1-5
 - RECEIVE 2-5, 2-6
 - SEND 2-7
- Trace Options panel Administrator Interface usage 4-44 to 4-53
- Transient Data queue initialization messages 5-5
- verifying installation 3-29
- cumulative statistics
 - collecting for CPT transactions 4-14
 - resetting using the Operator Control Menu panel 5-9

- customizing
 - CPT macros 3-9
 - CPT tools overview 2-3
 - RECEIVE tool 2-5
 - SEND tool 2-7

D

- data processing errors
 - RECEIVE tool 2-6
 - SEND tool 2-8
- DCT entries
 - copying from CPT SAMP library 3-14 to 3-15
- Destination Control Table (DCT) 2-5
- diagnostics
 - LISTEN tool 2-4
 - RECEIVE tool 2-6
 - SEND tool 2-9
- distributed configuration file 3-9
- document
 - audience 2
 - command notation and typographical conventions 3
 - references for additional information 2
 - where to find information (by chapter and appendix) 2
- Domain Name Resolver directory services 1-4

E

- endpoints
 - clearing cumulative statistic with the Operations Control Menu panel 4-58
 - confirming closed
 - for Listener tasks 4-57
 - with the Quiesce/Close Confirmation panel 4-58
 - displaying
 - data fields for in the Utilization Summary 4-9
 - details for using the Utilization Summary 4-11
 - inactive in the Utilization Summary 4-10
 - listener tasks 4-16
 - statistics with the Browse Detail panel 4-12, 4-14
 - UDP with the Browse Detail panel 4-17
 - terminating the CICS task with the Browse Detail panel 4-18
 - viewing active connections with the Browse Detail panel 4-12

error log

- changing queue size 4-60
- resetting queue size with the Operations Control Menu panel 4-59

Error Log queue

- browsing in the CPT Error Log panel 4-57
- determining size of 4-57

F

FTP

- copying PDS to disk 3-36
- installing CICS TD UNIX programs 3-36

G

Generate Trace Entry panel Administrator Interface usage 4-26 to 4-27

I

initialization

- failure
 - Task-Related User Exit 5-5
 - transient data queues not defined 5-5
- messages written in
 - CPT Transient Data queue 5-5
 - system console log 5-4
- SEND tool transaction and data transfer 2-7

initializing

- CPT
 - environment 1-10
 - Interface with the T09TSTRT program 5-4

installation

- SMP/E 3-4
- steps to install tape 3-4
- tape
 - loading the CNTL file from 3-4
 - steps to install 3-4

installation and maintenance

- SMP/E (overview) 1-9

Installation Verification Procedure (IVP)

- executing CICS 3-29
- for CPT 3-29

installing

- allocating CPT data sets 1-9
- CICS TD UNIX programs
 - test programs 3-31 to 3-34
 - with FTP 3-36

man pages 3-34

INSTT09 job, submitting to accept the product 3-7

L

LISTEN tool 2-4

- defining 2-4
- diagnostics 2-4
- initiating 2-4

Listen/Receive tool

- API trace options
 - setting 4-44
 - setting and removing 4-44
- modifying parameters 4-40
- starting with the Configuration Table parameters panel 4-39

Listen/Receive Tool panel

- Administrator Interface usage 4-35 to 4-39
- field descriptions 4-36 to 4-38

load module, T09CONFIG A-3

loopback mode, testing CPT 3-30

M

macro instructions

- T09MCICS 3-9
- T09MSEND 3-9

macros

- CPT configuration file 3-9
- defining T09MSEND for customizing SEND tool 2-7
- T09END A-3
- T09MCICS A-3
- T09MLSTN A-3
- T09MLSTN instructions 2-5
- T09MRECV A-3
- T09MRECV instructions 2-5
- T09MSEND A-3
 - to configure CPT A-3

man pages, installing 3-34

messages (by number) C-4 to C-125

- CPT252E C-62
- CPT280E C-65

MXT concurrent tasks, monitoring 3-28

O

Online Trace Options panel Administrator Interface usage 4-21 to 4-25

200
4030-200100

online tracing options, specifying using the Utilization Summary 4-11

Operations component Administrator Interface 4-54 to 4-61

Operations Control menu

- Administrator Interface usage 4-61
- starting the CPT Interface 4-61
- terminating CPT 5-6

Operations Control Menu panel

- Administrator Interface usage 4-54 to 4-55
- confirming CPT Interface termination 4-58
- diagnosing and managing the CPT Interface 4-54
- functions descriptions 4-54 to 4-55
- resetting error log queue size 4-59

Operator Control Menu panel, resetting cumulative statistics 5-9

P

parsing requirements

- RECEIVE tool 2-6
- writing data to the transient data queue 2-6

PLTPI table, defining CPT initialization program 3-13

PLTSD table, defining CPT termination program 3-13

PPT entries

- copying from CPT SAMP library 3-16 to 3-17, 3-18 to 3-20

processing

- errors
 - RECEIVE tool 2-6
 - SEND tool 2-8
 - SEND tool transport provider 2-8
- options for CPT tools 1-5

program statements

- CICS PCT 3-21, 3-25
- CICS PPT 3-16
- CPT PPT 3-18

programming interface facilities

- CPT API 1-8
- CPT tools 1-5
- Domain Name Resolver 1-4

Q

queue size

- changing online size 4-59
- resetting with the Operations Control Menu panel 4-59

Quiesce/Close Confirmation panel

- Administrator Interface usage 4-57
- confirming
 - closed endpoints for Listener tasks 4-57
 - endpoints are closed 4-58

Quiesce/Confirm panel

- aborting termination 4-18
- Administrator Interface usage 4-18

R

RECEIVE tool 2-5

- customizing 2-5
- data processing errors 2-6
- diagnostics 2-6
- parsing requirements 2-6
- processing errors 2-6
- reliability factors 2-6
- transactions and applications 2-5
- transport provider processing errors 2-6

reliability factors

- RECEIVE tool 2-6
- SEND tool 2-8

remote host, verifying activation 4-18

Reset Trace Online Queue panel

- Administrator Interface usage 4-28 to 4-29
- confirming the reset trace 4-28

Return Code Cross-Reference Table C-3

S

SEND tool 2-7

- control information 2-8
- customizing 2-7
- data processing error 2-8
- diagnostics 2-9
- initialization
 - automated transaction and transfer data 2-7
- processing errors 2-8
- reliability factors 2-8
- transport provider processing error 2-8

Send Tool panel

- Administrator Interface usage 4-45 to 4-47
- field descriptions 4-46 to 4-48

server transaction

- displaying
 - Utilization Summary details 4-11

server transactions

- displaying

- cumulative statistics using the Browse panel 4-14
 - detail with the Browse Detail panel 4-13
- SIT entries 3-28
- SMP/E
 - creating translation tables 3-39
 - installation 3-4
 - maintenance and installation (overview) 1-9
 - modifying translation tables 3-39
- software system requirements 3-3
- statements
 - CPT definition 3-13
 - resource definition 3-14
- statistics parameters
 - defining in the Configuration Table for utilization 4-9
- subroutines
 - CPT callable services overview 1-8
- system
 - console log initialization messages 5-4
 - software requirements 3-3

T

- T09CONFIG
 - configuration file 3-9
 - load module A-3
 - modifying 3-12
- T09END macro A-3
- T09MCICS
 - definition statement A-4
 - macro A-3
- T09MCICS macro instruction 3-9
- T09MEND definition statement A-18
- T09MLSTN
 - definition statement A-6
 - macro A-3
- T09MLSTN macro
 - instructions 2-5
- T09MRECV
 - definition statement A-11
 - macro A-3
- T09MRECV macro
 - instructions 2-5
- T09MSEND

macro A-3

- T09MSEND definition statement A-14
- T09MSEND macro
 - defining for customizing SEND tool 2-7
- T09MSEND macro instruction 3-9
- tar program, installing CICS TD UNIX test programs 3-31
- TCP/IP network programming interface overview 1-4
- TCPNAMES file, verifying 3-5
- terminating
 - CPT Interface 5-6, 5-7
- terminating CPT 1-10
- termination messages for CPT 5-7
- trace options
 - removing 4-20
 - setting with the Browse Detail panel 4-19
- translation tables
 - creating with SMP/E 3-39
 - customizing with SMP/E 3-39
- transport provider errors
 - RECEIVE tool 2-6
 - SEND tool 2-8
- typographic conventions used in the document 3

U

- utilization
 - Administrator
 - Interface 4-7 to 4-29
 - Interface panels 4-7
 - displaying CPT transaction summary statistics 4-9
 - online tracing 4-20
- Utilization Summary
 - browsing
 - client task details 4-11
 - server task details 4-11
 - displaying
 - endpoint data fields 4-9
 - endpoint details 4-11
 - inactive endpoints 4-10
 - listing CPT endpoints 4-9
 - specifying online tracing options 4-11
- Utilization Summary panel
 - Administrator Interface usage 4-7 to 4-11

2000-4030-200100

displaying endpoints in the Administrator Inter-
face 4-7

200801-024030-200100

Reader Comment Form

Please let us know if you find any errors in this document. We would also like to hear your comments, questions, or suggestions.

You can write to us at this address:

Interlink Computer Sciences, Inc.
Technical Publications Department
47370 Fremont Boulevard
Fremont, California 94538
U.S.A.

You can email us at this address:

techpubs@interlink.com

You can fax us at this number:

(510) 659-6381

If you have corrections or comments, please include this information:

- ◆ Document name or number (found on the front cover or the inner margin of each page)
- ◆ Page number

We appreciate your corrections and comments.

INTERLINK

Computer Sciences

Interlink Computer Sciences, Inc.

Corporate Headquarters
47370 Fremont Blvd.
Fremont, CA 94538
(510) 657-9800
(800) 422-3711
fax (510) 659-6381
email: info@interlink.com
web: <http://www.interlink.com>

For other U.S. sales offices
call (800) 422-3711

Canada
(403) 231-9800
fax (403) 266-6767

France
+33-1-45-079494
fax +33-1-45-079851

Germany

+49-2203-911853
fax +49-2203-911859

Spain

+34-1-352-62-65
fax +34-1-352-48-76

Switzerland

+41-26-460-85-50
fax +41-26-460-85-58

United Kingdom

+44-1923-77-6000
fax +44-1923-77-1100